This book is dedicated to OpenNA staff. Thanks, guys (no-gender)!!

--Gerhard Mourani

This book is printed on acid-free paper with 85% recycled content, 15% post-consumer waste. Open Network Architecture is commited to using paper with the highest recycled content available consistent with high quality.

Copyright © 2002 by Gerhard Mourani and Open Network Architecture, Inc.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted by Canada Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the copyright holders Gerhard Mourani and Open Network Architecture, Inc. 11090 Drouart, Montreal, PQ H3M 2S3, (514) 978-6183, fax (514) 333-0236. Requests to the Publisher for permission should be addressed to the Publishing Manager, at Open Network Architecture, Inc., E-mail: books@openna.com

This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. It is sold with the understanding that some grammatical mistakes could have occurred but this won't jeopardize the content or the issue raised herewith.

Title: Securing and Optimizing Linux: The Hacking Solution
Page Count: 1208
Version: 3.0
Last Revised: 2002-06-26
Publisher: Open Network Architecture, Inc.
Editor: Ted Nackad
Text Design & Drawings (Graphics): Bruno Mourani
Printing History: June 2000: First Publication.

Author's: Gerhard Mourani
Mail: gmourani@openna.com
Website: http://www.openna.com/

National Library Act. R.S., c. N-11, s. 1.

Legal Deposit, 2002

Securing and Optimizing Linux: The Hacking Solution / Open Network Architecture, Inc.
Published by Open Network Architecture, Inc., 11090 Drouart, Montreal, H3M 2S3, Canada.

Includes Index.
ISBN 0-9688793-1-4

Printed in Canada

Overview

Part I Installation Security
Chapter 1 Introduction
Chapter 2 Installation Issues

Part II System Security & Optimization
Chapter 3 General Security
Chapter 4 Pluggable Authentication Modules
Chapter 5 General Optimization
Chapter 6 Kernel Security & Optimization
Chapter 7 Process File System Management

Part III Network Security
Chapter 8 TCP/IP Network Management
Chapter 9 Firewall Basic Concept
Chapter 10 GIPTables Firewall
Chapter 11 Squid Proxy Server
Chapter 12 SquidGuard Filter
Chapter 13 FreeS/WAN VPN

Part IV Cryptography & Authentication
Chapter 14 GnuPG
Chapter 15 OpenSSL
Chapter 16 OpenSSH
Chapter 17 Sudo

Part V Monitoring & System Integrity
Chapter 18 sXid
Chapter 19 LogSentry
Chapter 20 HostSentry
Chapter 21 PortSentry
Chapter 22 Snort
Chapter 23 Tripwire

Part VI Super-Server
Chapter 24 UCSPI-TCP
Chapter 25 Xinetd

Part VII Management & Limitation
Chapter 26 NTP
Chapter 27 Quota

Part VIII Domain Name System & Dynamic Host Protocol
Chapter 28 ISC BIND & DNS
Chapter 29 ISC DHCP

Part IX Mail Transfer Agent Protocol
Chapter 30 Exim
Chapter 31 Qmail

Part X Internet Message Access Protocol
Chapter 32 tpop3d
Chapter 33 UW IMAP
Chapter 34 Qpopper

Part XI Anti-Spam & Anti-Virus
Chapter 35 SpamAssassin
Chapter 36 Sophos
Chapter 37 AMaViS

Part XII Database Server
Chapter 38 MySQL
Chapter 39 PostgreSQL
Chapter 40 OpenLDAP

Part XIII File Transfer Protocol
Chapter 41 ProFTPD
Chapter 42 vsFTPD

Part XIV Hypertext Transfer Protocol
Chapter 43 Apache
Chapter 44 PHP
Chapter 45 Mod_Perl

Part XV NetBios Protocol
Chapter 46 Samba

Part XVI Backup
Chapter 47 Tar & Dump

Part XVII Appendixes

Appendix A
Tweaks, Tips and Administration Tasks

Appendix B
Port list

Contents

Steps of installation 13
Author note 13
Audience 14
These installation instructions assume 15
Obtaining the example configuration files 15
Problem with Securing & Optimizing Linux 15
Acknowledgments 15

Introduction 19

What is Linux? 21
Some good reasons to use Linux 21
Let's dispel some of the fear, uncertainty, and doubt about Linux 21
Why choose pristine source? 22
Compiling software on your system 22
Build & install software on your system 23
Editing files with the `vi` editor tool 24
Recommended software to include in each type of servers 25

Installation Issues 29

Know your Hardware! 31
Creating the Linux Boot Disk 31
Beginning the installation of Linux 33
Installation Class and Method (Install Options) 34
Partition your system for Linux 35
Disk Partition (Manual Partitioning) 39
Selecting Package Groups 50
Boot Disk Creation 53
How to use `RPM` Commands 53
Starting and stopping daemon services 56
Software that must be uninstalled after installation of the server 57
Remove unnecessary documentation files 65
Remove unnecessary/empty files and directories 66
Software that must be installed after installation of the server 66

General Security 73

`BIOS` 75
Unplug your server from the network 75
Security as a policy 76
Choose a right password 76
The root account 77
Set login time out for the root account 77
Shell logging 78
The single-user login mode of Linux 79
Disabling `Ctrl-Alt-Delete` keyboard shutdown command 79
Limiting the default number of started `ttys` on the server 80
The `LILO` and `/etc/lilo.conf` file 80
The `GRUB` and `/boot/grub/grub.conf` file 82
The `/etc/services` file 84

The /etc/securetty file 85
Special accounts 85
Control mounting a file system 88
Mounting the /usr directory of Linux as read-only 89
Tighten scripts under /etc/init.d 91
Tighten scripts under /etc/cron.daily/ 91
Bits from root-owned programs 91
Don't let internal machines tell the server what their MAC address is 93
Unusual or hidden files 94
Finding Group and World Writable files and directories 95
Unowned files 96
Finding .rhosts files 96
Physical hard copies of all-important logs 97
Getting some more security by removing manual pages 99
System is compromised! 100

Pluggable Authentication Modules 101

The password length 103
Disabling console program access 105
Disabling all console access 105
The Login access control table 106
Tighten console permissions for privileged users 107
Putting limits on resource 109
Controlling access time to services 111
Blocking; su to root, by one and sundry 112
Using sudo instead of su for logging as super-user 113

General Optimization 116

Static vs. shared libraries 118
The Glibc 2.2 library of Linux 119
Why Linux programs are distributed as source 120
Some misunderstanding in the compiler flags options 121
The gcc specs file 122
Striping all binaries and libraries files 127
Tuning IDE Hard Disk Performance 128

Kernel Security & Optimization 133

Difference between a Modularized Kernel and a Monolithic Kernel 135
Making an emergency boot floppy 138
Preparing the Kernel for the installation 139
Applying the Grsecurity kernel patch 141
Obtaining and Installing Grsecurity 141
Tuning the Kernel 142
Cleaning up the Kernel 143
Configuring the Kernel 145
Compiling the Kernel 190
Installing the Kernel 190
Verifying or upgrading your boot loader 192
Reconfiguring /etc/modules.conf file 194
Rebooting your system to load the new kernel 195
Delete programs, edit files pertaining to modules 195

Making a new rescue floppy for Modularized Kernel 196
Making a emergency boot floppy disk for Monolithic Kernel 196

Process file system management 199

What is `sysctl`? 202
`/proc/sys/vm`: The virtual memory subsystem of Linux 202
`/proc/sys/fs`: The file system data of Linux 209
`/proc/sys/net/ipv4`: IPV4 settings of Linux 211
Other possible optimization of the system 219

TCP/IP Network Management 225

`TCP/IP` security problem overview 228
Installing more than one Ethernet Card per Machine 232
Files-Networking Functionality 233
Testing `TCP/IP` Networking 237
The last checkup 240

Firewall Basic Concept 241

What is the IANA? 243
The ports numbers 243
What is a Firewall? 245
Packet Filter vs. Application Gateway 245
What is a Network Firewall Security Policy? 247
The Demilitarized Zone 248
Linux `IPTables` Firewall Packet Filter 249
The Netfilter Architecture 249

`GIPTables` Firewall 255

Building a kernel with `IPTables` support 259
Compiling - Optimizing & Installing `GIPTables` 262
Configuring `GIPTables` 263
`/etc/giptables.conf`: The `GIPTables` Configuration File 263
`/etc/rc.d/rc.giptables.blocked`: The `GIPTables` Blocked File 274
`/etc/init.d/giptables`: The `GIPTables` Initialization File 275
The `GIPTables` Firewall Module Files 276
How `GIPTables` parameters work? 277
Running the type of `GIPTables` firewall that you need 283
The `GIPTables` configuration file for a Gateway/Proxy Server 284
`GIPTables-Firewall` Administrative Tools 302

Squid Proxy Server 305

Compiling - Optimizing & Installing `Squid` 309
Configuring `Squid` 313
Running `Squid` with Users Authentication Support 326
Securing `Squid` 330
Optimizing `Squid` 333
`Squid` Administrative Tools 333
The `cachemgr.cgi` program utility of `Squid` 335

SquidGuard Filter 337

Compiling - Optimizing & Installing SquidGuard 340
Configuring SquidGuard 342
Testing SquidGuard 350
Optimizing SquidGuard 351

FreeS/WAN VPN 355

Compiling - Optimizing & Installing FreeS/WAN 360
Configuring FreeS/WAN 363
Configuring RSA private keys secrets 367
Requiring network setup for IPSec 372
Testing the FreeS/WAN installation 374

GnuPG 379

Compiling - Optimizing & Installing GnuPG 382
Using GnuPG under Linux terminal 384

OpenSSL 391

Compiling - Optimizing & Installing OpenSSL 396
Configuring OpenSSL 398
OpenSSL Administrative Tools 404
Securing OpenSSL 409

OpenSSH 411

Compiling - Optimizing & Installing OpenSSH 414
Configuring OpenSSH 417
Running OpenSSH in a chroot jail 427
Creating OpenSSH private & public keys 432
OpenSSH Users Tools 434

Sudo 437

Compiling - Optimizing & Installing Sudo 440
Configuring Sudo 442
A more complex sudoers configuration file 444
Securing Sudo 447
Sudo Users Tools 447

sXid 451

Compiling - Optimizing & Installing sXid 454
Configuring sXid 455
sXid Administrative Tools 457

LogSentry 459

Compiling - Optimizing & Installing LogSentry 462
Configuring LogSentry 466

HostSentry 467

Compiling - Optimizing & Installing HostSentry 470
Configuring HostSentry 474

PortSentry 481

Compiling - Optimizing & Installing PortSentry 484
Configuring PortSentry 487
Removing hosts that have been blocked by PortSentry 494

Snort 495

Compiling - Optimizing & Installing Snort 499
Configuring Snort 501
Running Snort in a chroot jail 507

Tripwire 511

Compiling - Optimizing & Installing Tripwire 514
Configuring Tripwire 517
Running Tripwire for the first time 526
Securing Tripwire 528
Tripwire Administrative Tools 528

ucspi-tcp 533

Compiling - Optimizing & Installing ucsip-tcp 536
Using ucsip-tcp 538

Xinetd 541

Compiling - Optimizing & Installing Xinetd 544
Configuring Xinetd 546
The /etc/xinetd.d directory 547

NTP 559

Compiling - Optimizing & Installing NTP 564
Configuring NTP 566
Running NTP in Client Mode 566
Running NTP in Server Mode 572
Running NTP in a chroot jail 574
NTP Administrative Tools 578

Quota 581

Build a kernel with Quota support enable 584
Compiling - Optimizing & Installing Quota 584

Modifying the /etc/fstab file 586
Creating the aquota.user and aquota.group files 587
Assigning Quota for Users and Groups 587
Quota Administrative Tools 590

ISC BIND & DNS 593

Compiling - Optimizing & Installing ISC BIND & DNS 598
Configuring ISC BIND & DNS 600
Running ISC BIND & DNS as Caching-Only Name Server 601
Running ISC BIND & DNS as Primary Master Name Server 610
Running ISC BIND & DNS as Secondary Slave Name Server 615
Running ISC BIND & DNS in a chroot jail 617
Securing ISC BIND & DNS 621
Optimizing ISC BIND & DNS 638
ISC BIND & DNS Administrative Tools 641
ISC BIND & DNS Users Tools 643

ISC DHCP 645

Building a kernel with ISC DHCP support 649
Compiling - Optimizing & Installing ISC DHCP 650
Configuring ISC DHCP 654
Testing the DHCP server 662
Running ISC DHCP in a chroot jail 664
Securing ISC DHCP 675
Running the DHCP client for Linux 676

Exim 683

Compiling - Optimizing & Installing Exim 688
Configuring Exim 693
Testing Exim 716
Allowing Users to authenticate with Exim before relaying 719
Running Exim with SSL support 722
Running Exim with Virtual Hosts support 729
Running Exim with Maildir support 732
Running Exim with mail quota support 734
Running Exim as a Null Client Mail Server 735
Exim Administrative Tools 738

Qmail 741

Compiling, Optimizing & Installing Qmail 745
Configuring Qmail 751
Testing Qmail 755
Allowing Users to authenticate with Qmail before relaying 756
Running Qmail with SSL support 760
Running Qmail with Virtual Hosts support 765
Running Qmail as a Null Client Mail Server 769
Running Qmail as a Mini-Qmail Mail Server 773
Running qmail-pop3d with SSL support 777

Qmail Administrative Tools 780
Qmail Users Tools 781

tpop3d 785

Compiling - Optimizing & Installing tpop3d 790
Configuring tpop3d 791
Securing tpop3d 795

UW IMAP 797

Compiling - Optimizing & Installing UW IMAP 801
Configuring UW IMAP 805
Enable IMAP or POP services via UCSPI-TCP 807
Enable IMAP or POP services via Xinetd 808
Securing UW IMAP 810
Running UW IMAP with SSL support 811

Qpopper 815

Compiling - Optimizing & Installing Qpopper 819
Configuring Qpopper 821
Securing Qpopper 825
Running Qpopper with SSL support 827

SpamAssassin 835

Compiling - Optimizing & Installing SpamAssassin 839
Configuring SpamAssassin 840
Testing SpamAssassin 842
Running SpamAssassin with Exim 843
Running SpamAssassin with Qmail 844

Sophos 849

Compiling & Installing Sophos 853
Configuring Sophos 854
Testing Sophos 855

AMaViS 857

Verifying & installing all the additional prerequisites to run AMaViS 860
Compiling - Optimizing & Installing AMaViS 872
Running AMaViS with Exim 875
Running AMaViS with Qmail 877
Testing AMaViS 878

MySQL 881

Compiling - Optimizing & Installing MySQL 886
Configuring MySQL 888
Securing MySQL 893

Optimizing MySQL 894
MySQL Administrative Tools 899

PostgreSQL 907

Compiling - Optimizing & Installing PostgreSQL 910
Configuring PostgreSQL 913
Running PostgreSQL with SSL support 918
Securing PostgreSQL 924
Optimizing PostgreSQL 928
PostgreSQL Administrative Tools 929

OpenLDAP 935

Compiling - Optimizing & Installing OpenLDAP 940
Configuring OpenLDAP 945
Running OpenLDAP with TLS/SSL support 950
Running OpenLDAP in a chroot jail 954
Securing OpenLDAP 961
Optimizing OpenLDAP 962
OpenLDAP Administrative Tools 963
OpenLDAP Users Tools 967

ProFTPD 971

Compiling - Optimizing & Installing ProFTPD 976
Configuring ProFTPD 980
Creating an account for FTP client to connect to the FTP server 992
Setup an anonymous FTP server 993
Allow anonymous users to upload to the FTP server 997
Running ProFTPD with SSL support 1000
Securing ProFTPD 1005
ProFTPD Administrative Tools 1006

vsFTPd 1009

Compiling - Optimizing & Installing vsFTPd 1014
Configuring vsFTPd 1015
Creating an account for FTP client to connect to the FTP server 1021
Setup an anonymous FTP server 1022
Allow anonymous users to upload to the FTP server 1024

Apache 1029

Compiling - Optimizing & Installing Apache 1034
Configuring Apache 1040
Running Apache with TLS/SSL support 1051
Running Apache in a chroot jail 1055
Running Apache with users authentication support 1063
Caching frequently requested static files 1065
Some statistics about Apache and Linux 1066

PHP **1069**

Compiling - Optimizing & Installing PHP 1073
Configuring PHP 1076
Running PHP in a chroot jail 1084
Running PHP with the PHP Accelerator program 1085

Mod_Perl **1089**

Compiling - Optimizing & Installing Mod_Perl 1093
Configuring Mod_Perl 1094
Running Mod_Perl in a chroot jail 1095

Samba **1099**

Compiling - Optimizing & Installing Samba 1104
Configuring Samba 1106
Running Samba with TLS/SSL support 1116
Securing Samba 1121
Optimizing Samba 1123
Samba Administrative Tools 1125
Samba Users Tools 1126

Tar & Dump **1129**

The tar backup program 1131
Making backups with tar 1132
Automating tasks of backups made with tar 1134
Restoring files with tar 1136
The dump backup program 1138
Making backups with dump 1139
Restoring files with dump 1141
Backing up and restoring over the network 1143

APPENDIX A **1151**

APPENDIX B **1157**

Steps of installation

Depending of your level of knowledge in Linux, you can read this book from the beginning through to the end of the chapters that interest you. Each chapter and section of this book appears in a manner that lets you read only the parts of your interest without the need to schedule one day of reading. Too many books on the market take myriad pages to explain something that can be explained in two lines, I'm sure that a lot of you agree with my opinion. This book tries to be different by talking about only the essential and important information that the readers want to know by eliminating all the nonsense.

Although you can read this book in the order you want, there is a particular order that you could follow if something seems to be confusing you. The steps shown below are what I recommend:

- ✓ Setup Linux in your computer.
- ✓ Remove all the unnecessary `RPM`'s packages.
- ✓ Install the necessary `RPM`'s packages for compilation of software (if needed).
- ✓ Secure the system in general.
- ✓ Optimize the system in general.
- ✓ Reinstall, recompile and customize the Kernel to fit your specific system.
- ✓ Configure firewall script according to which services will be installed in your system.
- ✓ Install `OpenSSL` to be able to use encryption with the Linux server.
- ✓ Install `OpenSSH` to be able to make secure remote administration tasks.
- ✓ Install `Sudo`.
- ✓ Install `sXid`.
- ✓ Install `LogSentry`.
- ✓ Install `PortSentry`.
- ✓ Install `Tripwire`.
- ✓ Install `ICS BIND/DNS`.
- ✓ Install `Exim` or `Qmail`.
- ✓ Install any software you need after to enable specific services into the server.

Author note

According to some surveys on the Internet, Linux will be the number one operating system for a server platform in year 2003. Presently it is number two and no one at one time thought that it would be in this second place. Many organizations, companies, universities, governments, and the military, etc, kept quiet about it. Crackers use it as the operating system by excellence to crack computers around the world. Why do so many people use it instead of other well know operating systems? The answer is simple, Linux is free and the most powerful, reliable, and secure operating system in the world, providing it is well configured. Millions of programmers, home users, hackers, developers, etc work to develop on a voluntary basis, different programs related to security, services, and share their work with other people to improve it without expecting anything in return. This is the revolution of the Open Source movement that we see and hear about so often on the Internet and in the media.

If crackers can use Linux to penetrate servers, security specialists can use the same means to protect servers (to win a war, you should at least have equivalent weapons to what your enemy may be using). When security holes are encountered, Linux is the one operating system that has a solution and that is not by chance. Now someone may say: with all these beautiful features why is Linux not as popular as other well know operating system? There are many reasons and different answers on the Internet. I would just say that like everything else in life, anything that we are to expect the most of, is more difficult to get than the average and easier to acquire. Linux and *NIX are more difficult to learn than any other operating system. It is only for those who want to know computers in depth and know what they doing. People prefer to use other OS's, which are easy to operate but hard to understand what is happening in the background since they only have to click on a button without really knowing what their actions imply. Every UNIX operating system like Linux will lead you unconsciously to know exactly what you are doing because if you pursue without understanding what is happening by the decision you made, then nothing will surely work as expected. This is why with Linux; you will know the real meaning of a computer and especially a server environment where every decision warrants an action which will closely impact on the security of your organization and employees.

Many Web sites are open to all sorts of "web hacking." According to the Computer Security Institute and the FBI's joint survey, 90% of 643 computer security practitioners from government agencies, private corporations, and universities detected cyber attacks last year. Over $265,589,940 in financial losses was reported by 273 organizations.

Many readers of the previous version of this book told me that the book was an easy step by step guide for newbie's, I am flattered but I prefer to admit that it was targeting for a technical audience and I assumed the reader had some background in Linux, UNIX systems. If this is not true in your case, I highly recommend you to read some good books in network administration related to UNIX and especially to Linux before venturing into this book. Remember talking about security and optimization is a very serious endeavor. It is very important to be attentive and understand every detail in this book and if difficulties arise, try to go back and reread the explanation will save a lot of frustration. Once again, security is not a game and crackers await only one single error from your part to enter your system. A castle has many doors and if just one stays open, will be enough to let intruders into your fortress. You have been warned.

Many efforts went into the making of this book, making sure that the results were as accurate as possible. If you find any abnormalities, inconsistent results, errors, omissions or anything else that doesn't look right, please let me know so I can investigate the problem and/or correct the error. Suggestions for future versions are also welcome and appreciated. A web site dedicated to this book is available on the Internet for your convenience. If you any have problem, question, recommendation, etc, please go to the following URL: http://www.openna.com/. We made this site for you.

Audience

This book is intended for a technical audience and system administrators who manage Linux servers, but it also includes material for home users and others. It discusses how to install and setup a Linux server with all the necessary security and optimization for a high performance Linux specific machine. It can also be applied with some minor changes to other Linux variants without difficulty. Since we speak of optimization and security configuration, we will use a source distribution (`tar.gz`) program for critical server software like `Apache`, `ISC BIND/DNS`, `Samba`, `Squid`, `OpenSSL` etc. Source packages give us fast upgrades; security updates when necessary, and better compilation, customization, and optimization options for specific machines that often aren't available with `RPM` packages.

These installation instructions assume

You have a CD-ROM drive on your computer and the Official Red Hat Linux or OpenNA Linux CD-ROM. Installations were tested on the Official Red Hat Linux version 7.3 and OpenNA Linux.

You should familiarize yourself with the hardware on which the operating system will be installed. After examining the hardware, the rest of this document guides you, step-by-step, through the installation process.

Obtaining the example configuration files

In a true server environment and especially when Graphical User Interface is not installed, we will often use text files, scripts, shell, etc. Throughout this book we will see shell commands, script files, configuration files and many other actions to execute on the terminal of the server. You can enter them manually or use the compressed archive file that I made which contains all configuration examples and paste them directly to your terminal. This seems to be useful in many cases to save time.

The example configuration files in this book are available electronically via HTTP from this URL: http://www.openna.com/products/books/securing-optimizing-linux/3rdedition/index.htm

- In either case, extract the files into your Linux server from the archive by typing:

```
[root@deep /]# cd /var/tmp
[root@deep tmp]# tar xzpf floppy-3.0.tgz
```

If you cannot get the examples from the Internet, please contact the author at this email address: gmourani@openna.com

Problem with Securing & Optimizing Linux

When you encounter a problem in "Securing & Optimizing Linux" we want to hear about it. Your reports are an important part in making the book more reliable, because even with the utmost care we cannot guarantee that every part of the book will work on every platform under every circumstance.

We cannot promise to fix every error right away. If the problem is obvious, critical, or affects a lot of users, chances are that someone will look into it. It could also happen that we tell you to update to a newer version to see if the problem persists there. Or we might decide that the problem cannot be fixed until some major rewriting has been done. If you need help immediately, consider obtaining a commercial support contract or try our Q&A archive from the mailing list for an answer.

Below are some important links:

OpenNA web site: http://www.openna.com/
Mailing list: http://www.openna.com/support/mailing/
Support: http://www.openna.com/support/
RPM Download: http://www.openna.com/download/

Acknowledgments

I would like to thank all the OpenNA staff for their hard works and patience. A special gratitude and many thanks to Colin Henry who made tremendous efforts to make this book grammatically and orthographically sound in a professional manner. Adrian Pascalau for its time and help in the open source community and all Linux users around the world who have participated by providing good comments, ideas, recommendations and suggestions.



Part I

Installation Security

CHAPTER 1

Introduction

IN THIS CHAPTER

1. **What is Linux?**
2. **Some good reasons to use Linux**
3. **Let's dispel some of the fear, uncertainty, and doubt about Linux**
4. **Why choose Pristine source?**
5. **Compiling software on your system**
6. **Build, Install software on your system**
7. **Editing files with the `vi` editor tool**
8. **Recommended software to include in each type of servers**

Introduction

What is Linux?

Linux is an operating system that was first created at the University of Helsinki in Finland by a young student named Linus Torvalds. At this time the student was working on a UNIX system that was running on an expensive platform. Because of his low budget, and his need to work at home, he decided to create a copy of the UNIX system in order to run it on a less expensive platform, such as an IBM PC. He began his work in 1991 when he released version 0.02 and worked steadily until 1994 when version 1.0 of the Linux Kernel was released.

The Linux operating system is developed under the GNU **G**eneral **P**ublic **L**icense (also known as GNU GPL) and its source code is freely available to everyone who downloads it via the Internet. The CD-ROM version of Linux is also available in many stores, and companies that provide it will charge you for the cost of the media and support. Linux may be used for a wide variety of purposes including networking, software development, and as an end-user platform. Linux is often considered an excellent, low-cost alternative to other more expensive operating systems because you can install it on multiple computers without paying more.

Some good reasons to use Linux

There are no royalty or licensing fees for using Linux and the source code can be modified to fit your needs. The results can be sold for profit, but the original authors retain copyright and you must provide the source to your modifications.

Because it comes with source code to the kernel, it is quite portable. Linux runs on more CPUs and platforms than any other computer operating system.

The recent direction of the software and hardware industry is to push consumers to purchase faster computers with more system memory and hard drive storage. Linux systems are not affected by those industries' orientation because of its capacity to run on any kind of computer, even aging x486-based computers with limited amounts of RAM.

Linux is a true multi-tasking operating system similar to its brother, UNIX. It uses sophisticated, state-of-the-art memory management techniques to control all system processes. That means that if a program crashes you can kill it and continue working with confidence.

Another benefit is that Linux is practically immunized against all kinds of viruses that we find in other operating systems. To date we have found only two viruses that were effective on Linux systems - well, actually they are Trojan Horses.

Let's dispel some of the fear, uncertainty, and doubt about Linux

It's a toy operating system
Fortune 500 companies, governments, and consumers more and more use Linux as a cost-effective computing solution. It has been used, and is still used, by big companies like IBM, Amtrak, NASA, and others.

There's no support
Every Linux distribution comes with more than 12,000 pages of documentation. Commercial Linux distributions offer initial support for registered users, and small business and corporate accounts can get 24/7 supports through a number of commercial support companies. As an Open Source operating system, there's no six-month wait for a service release, plus the online Linux community fixes many serious bugs within hours.

Why choose pristine source?

All the programs in Red Hat and OpenNA distributions of Linux are provided as `RPM` files. An `RPM` file, also known, as a “package”, is a way of distributing software so that it can be easily installed, upgraded, queried, and deleted. However, in the Unix world, the defacto-standard for package distribution continues to be by way of so-called “tarballs”. Tarballs are simply compressed files that can be readable and uncompressed with the “`tar`” utility. Installing from `tar` is usually significantly more tedious than using `RPM`. So why would we choose to do so?

1) Unfortunately, it takes a few weeks for developers and helpers to get the latest version of a package converted to `RPM`’s because many developers first release them as tarballs.

2) When developers and vendors release a new `RPM`, they include a lot of options that often aren’t necessary. Those organizations and companies don’t know what options you will need and what you will not, so they include the most used to fit the needs of everyone.

3) Often `RPM`s are not optimized for your specific processors; companies like Red Hat Linux build `RPM`’s based on a standard PC. This permits their `RPM` packages to be installed on all sorts of computers since compiling a program for an i386 machine means it will work on all systems.

4) Sometimes you download and install `RPM`’s, which other people around the world are building and make available for you to use. This can pose conflicts in certain cases depending how this individual built the package, such as errors, security and all the other problems described above.

Compiling software on your system

A program is something a computer can execute. Originally, somebody wrote the "source code" in a programming language he/she could understand (e.g., C, C++). The program "source code" also makes sense to a compiler that converts the instructions into a binary file suited to whatever processor is wanted (e.g. a 386 or similar). A modern file format for these "executable" programs is `ELF`. The programmer compiles his source code on the compiler and gets a result of some sort. It's not at all uncommon that early attempts fail to compile, or having compiled, fail to act as expected. Half of programming is tracking down and fixing these problems (debugging).

For the beginners there are more aspect and new words relating to the compilation of source code that you must know, these include but are not limited to:

Multiple Files (Linking)

One-file programs are quite rare. Usually there are a number of files (say `*.c`, `*.cpp`, etc) that are each compiled into object files (`*.o`) and then linked into an executable. The compiler is usually used to perform the linking and calls the '`ld`' program behind the scenes.

Makefiles

Makefiles are intended to aid you in building your program the same way each time. They also often help with increasing the speed of a program. The “`make`” program uses “dependencies” in the `Makefile` to decide what parts of the program need to be recompiled. If you change one source file out of fifty you hope to get away with one compile and one link step, instead of starting from scratch.

Libraries
Programs can be linked not only to object files (`*.o`) but also to libraries that are collections of object files. There are two forms of linking to libraries: static, where the code goes in the executable file, and dynamic, where the code is collected when the program starts to run.

Patches
It was common for executable files to be given corrections without recompiling them. Now this practice has died out; in modern days, people change a small portion of the source code, putting a change into a file called a "patch". Where different versions of a program are required, small changes to code can be released this way, saving the trouble of having two large distributions.

Errors in Compilation and Linking
Errors in compilation and linking are often due to typos, omissions, or misuse of the language. You have to check that the right "includes file" is used for the functions you are calling. Unreferenced symbols are the sign of an incomplete link step. Also check if the necessary development libraries (`GLIBC`) or tools (`GCC`, `DEV86`, `MAKE`, etc) are installed on your system.

Debugging
Debugging is a large topic. It usually helps to have statements in the code that inform you of what is happening. To avoid drowning in output you might sometimes get them to print out only the first 3 passes in a loop. Checking that variables have passed correctly between modules often helps. Get familiar with your debugging tools.

Build & install software on your system

You will see in this book that we use many different compile commands to build and install programs on the server. These commands are UNIX compatible and are used on all variants of *NIX machines to compile and install software.

The procedures to compile and install software tarballs on your server are as follows:

1. First of all, you must download the tarball from your trusted software archive site. Usually from the main site of the software you hope to install.

2. After downloading the tarball, change to the `/var/tmp` directory (note that other paths are possible, at personal discretion) and untar the archive by typing the commands (as root) as in the following example:

```
[root@deep /]# tar xzpf foo.tar.gz
```

The above command will extract all files from the example `foo.tar.gz` compressed archive and will create a new directory with the name of the software from the path where you executed the command.

The "`x`" option tells `tar` to extract all files from the archive.
The "`z`" option tells `tar` that the archive is compressed with `gzip` utility.
The "`p`" option maintains the original permissions the files had when the archive was created.
The "`f`" option tells `tar` that the very next argument is the file name.

Once the tarball has been decompressed into the appropriate directory, you will almost certainly find a "README" and/or an "INSTALL" file included with the newly decompressed files, with further instructions on how to prepare the software package for use. Likely, you will need to enter commands similar to the following example:

```
./configure
make
make install
```

The above commands, **./configure** will configure the software to ensure your system has the necessary libraries to successfully compile the package, **make** will compile all the source files into executable binaries. Finally, **make install** will install the binaries and any supporting files into the appropriate locations. Other specific commands that you'll see in this book for compilation and installation procedure will be:

```
make depend
strip
chown
```

The **make depend** command will build and make the necessary dependencies for different files. The **strip** command will discard all symbols from the object files. This means that our binary file will be smaller in size. This will improve the performance of the program, since there will be fewer lines to read by the system when it executes the binary. The **chown** command will set the correct file owner and group permissions for the binaries. More commands will be explained in the sections concerning program installation.

Editing files with the vi editor tool

The vi program is a text editor that you can use to edit any text and particularly programs. During installation of software, the user will often have to edit text files, like Makefiles or configuration files. The following are some of the more important keystroke commands to get around in vi. I decided to introduce the vi commands now since it is necessary to use vi throughout this book.

Command	Result
i	Notifies vi to insert text before the cursor
a	Notifies vi to append text after the cursor
dd	Notifies vi to delete the current line
x	Notifies vi to delete the current character
Esc	Notifies vi to end the insert or append mode
u	Notifies vi to undo the last command
Ctrl+f	Scroll up one page
Ctrl+b	Scroll down one page
/string	Search forward for string
:f	Display filename and current line number
:q	Quit editor
:q!	Quit editor without saving changes
:wq	Save changes and exit editor

Recommended software to include in each type of servers

If you buy binaries, you will not get any equity and ownership of source code. Source code is a very valuable asset and binaries have no value. Buying software may become a thing of the past. You only need to buy good hardware; it is worth spending money on the hardware and gets the software from the Internet. The important point is that it is the computer hardware that is doing the bulk of the work. The hardware is the real workhorse and the software is just driving it. It is for this reason that we believe in working with and using Open source software. Much of the software and services that come with Linux are open source and allow the user to use and modify them in an undiscriminating way according to the **G**eneral **P**ublic **L**icense.

Linux has quickly become the most practical and friendly used platform for e-business -- and with good reason. Linux offers users stability, functionality and value that rivals any platform in the industry. Millions of users worldwide have chosen Linux for running their applications, from web and email servers to departmental and enterprise vertical application servers. To respond to your needs and to let you know how you can share services between systems I have developed ten different types of servers, which cover the majority of servers' functions and enterprise demands.

Often companies try to centralize many services into one server to save money, it is well known and often seen that there are conflicts between the technical departments and purchasing agents of companies about investment and expenditure when it comes to buying new equipment. When we consider security and optimization, it is of the utmost importance not to run too many services on one server, it is highly recommended to distribute tasks and services between multiple systems. The table below shows you which software and services we recommend to for each type of Linux server.

The following conventions will explain the interpretations of these tables:

- **Optional Components**: components that may be included to improve the features of the server or to fit special requirements.
- **Security Software Required**: what we consider as minimum-security software to have installed on the server to improve security.
- **Security Software Recommended**: what we recommend for the optimal security of the servers.

Mail Server	Web Server	Gateway Server
Exim or Qmail (SMTP Server) BIND/DNS (Caching) IPTables Firewall GIPTables ---------- IMAP/POP only for Exim	Apache Qmail BIND/DNS (Caching) IPTables Firewall GIPTables	BIND/DNS (Caching) Qmail IPTables Firewall GIPTables ---------- Squid SuidGuard
Optional Components	Optional Components	Optional Components
	Mod_PHP Mod_SSL Mod-Perl	DHCP
Security Software Required	Security Software Required	Security Software Required
Grsecurity OpenSSL OpenSSH Tripwire Sudo	Grsecurity OpenSSL OpenSSH Tripwire Sudo	Grsecurity OpenSSL OpenSSH Tripwire Sudo
Security Software recommended	Security Software recommended	Security Software recommended
GnuPG sXid Logcheck HostSentry PortSentry	GnuPG sXid Logcheck HostSentry PortSentry	GnuPG sXid Logcheck HostSentry PortSentry

FTP Server	Domain Name Server	File Sharing Server
ProFTPD Qmail BIND/DNS (Caching) IPTables Firewall GIPTables	Primary BIND/DNS (Server) Qmail IPTables Firewall GIPTables ---------- Secondary BIND/DNS (Server)	Samba Qmail BIND/DNS (Caching) IPTables Firewall GIPTables
Optional Components	Optional Components	Optional Components
Anonymous FTP (Server)		
Security Software Required	Security Software Required	Security Software Required
Grsecurity OpenSSL OpenSSH Tripwire Sudo	Grsecurity OpenSSL OpenSSH Tripwire Sudo	Grsecurity OpenSSL OpenSSH Tripwire Sudo
Security Software recommended	Security Software recommended	Security Software recommended
GnuPG sXid Logcheck HostSentry PortSentry	GnuPG sXid Logcheck HostSentry PortSentry	GnuPG sXid Logcheck HostSentry PortSentry

Database server	Backup server	VPN Server
PostgreSQL (Client & Server) Qmail BIND/DNS (Caching) IPTables Firewall GIPTables ---------- MySQL (Client & Server) ---------- OpenLDAP (Client & Servers)	Amanda Qmail BIND/DNS (Caching) Dump Utility IPTables Firewall GIPTables	FreeS/WAN VPN (Server) Qmail BIND/DNS (Caching) IPTables Firewall GIPTables
Optional Components	Optional Components	Optional Components
Security Software Required	Security Software Required	Security Software Required
Grsecurity OpenSSL OpenSSH Tripwire Sudo	Grsecurity OpenSSL OpenSSH Tripwire Sudo	Grsecurity OpenSSL OpenSSH Tripwire Sudo
Security Software recommended	Security Software recommended	Security Software recommended
GnuPG sXid Logcheck HostSentry PortSentry	GnuPG sXid Logcheck HostSentry PortSentry	GnuPG sXid Logcheck HostSentry PortSentry

CHAPTER 2

Installation Issues

IN THIS CHAPTER

1. **Know your Hardware!**
2. **Creating the Linux Boot Disk**
3. **Beginning the installation of Linux**
4. **Installation Class and Method (Install Options)**
5. **Partition your system for Linux**
6. **Disk Partition (Manual Partitioning)**
7. **Selecting Package Groups**
8. **Boot Disk Creation**
9. **How to use `RPM` Commands**
10. **Starting and stopping daemon services**
11. **Software that must be uninstalled after installation of the server**
12. **Remove unnecessary documentation files**
13. **Remove unnecessary/empty files and directories**
14. **Software that must be installed after installation of the server**

Linux Installation

Abstract
This part of the book deals with the basic knowledge required to properly install a Linux OS, in our case this is going to be Red Hat Linux, on your system in the most secure and clean manner available.

We have structured this chapter in a manner that follows the original installation of the Red Hat Linux operating system from CD-ROM. Each section below refers to, and will guide you through, the different screens that appear during the setup of your system after booting from the Red Hat boot diskette. We promise that it will be interesting to have the machine you want to install Linux on ready and near you when you follow the steps described below.

You will see that through the beginning of the installation of Linux, there are many options, parameters, and hacks that you can set before the system boots up for the first time.

Know your Hardware!
Understanding the hardware of your computer is essential for a successful installation of Linux. Therefore, you should take a moment and familiarize yourself with your computer hardware. Be prepared to answer the following questions:

1. How many hard drives do you have?
2. What size is each hard drive (eg, 15GB)?
3. If you have more than one hard drive, which is the primary one?
4. What kind of hard drive do you have (eg, IDE ATA/66, SCSI)?
5. How much RAM do you have (eg, 256MB RAM)?
6. Do you have a SCSI adapter? If so, who made it and what model is it?
7. Do you have a RAID system? If so, who made it and what model is it?
8. What type of mouse do you have (eg, PS/2, Microsoft, Logitech)?
9. How many buttons does your mouse have (2/3)?
10. If you have a serial mouse, what COM port is it connected to (eg, COM1)?
11. What is the make and model of your video card? How much video RAM do you have (eg, 8MB)?
12. What kind of monitor do you have (make and model)?
13. Will you be connected to a network? If so, what will be the following:
 a. Your IP address?
 b. Your netmask?
 c. Your gateway address?
 d. Your domain name server's IP address?
 e. Your domain name?
 f. Your hostname?
 g. Your types of network(s) card(s) (makes and model)?
 h. Your number of card(s) (makes and model)?

Creating the Linux Boot Disk
The first thing to do is to create an installation diskette, also known as a boot disk. If you have purchased the official Red Hat Linux CD-ROM, you will find a floppy disk called "Boot Diskette" in the Red Hat Linux box so you don't need to create it.

Sometimes, you may find that the installation will fail using the standard diskette image that comes with the official Red Hat Linux CD-ROM. If this happens, a revised diskette is required in order for the installation to work properly. In these cases, special images are available via the Red Hat Linux Errata web page to solve the problem (http://www.redhat.com/errata).

Since this, is a relatively rare occurrence, you will save time if you try to use the standard diskette images first, and then review the Errata only if you experience any problems completing the installation. Below, we will show you two methods to create the installation Boot Disk, the first method is to use an existing Microsoft Windows computer and the second using an existing Linux computer.

Making a Diskette under MS-DOS:

Before you make the boot disk, insert the Official Red Hat Linux CD-ROM Disk 1 in your computer that runs the Windows operating system. When the program asks for the filename, enter **`boot.img`** for the boot disk. To make the floppies under MS-DOS, you need to use these commands (assuming your CD-ROM is drive D: and contain the Official Red Hat Linux CD-ROM).

- Open the Command Prompt under Windows: Start | Programs | Command Prompt

```
C:\> d:
D:\> cd \dosutils
D:\dosutils> rawrite
Enter disk image source file name: ..\images\boot.img
Enter target diskette drive: a:
Please insert a formatted diskette into drive A: and press -ENTER- :

D:\dosutils>exit
```

The `rawrite.exe` program asks for the filename of the disk image: Enter **`boot.img`** and insert a blank floppy into drive A. It will then ask for a disk to write to: Enter `a:`, and when complete, label the disk “Red Hat boot disk”, for example.

Making a Diskette under a Linux-Like OS:

To make a diskette under Linux or any other variant of Linux-Like operating system, you must have permission to write to the device representing the floppy drive (known as `/dev/fd0H1440` under Linux).

This permission is granted when you log in to the system as the super-user “`root`”. Once you have logged as “`root`”, insert a blank formatted diskette into the diskette drive of your computer without issuing a `mount` command on it. Now it’s time to mount the Red Hat Linux CD-ROM on Linux and change to the directory containing the desired image file to create the boot disk.

- Insert a blank formatted diskette into the diskette drive
 Insert the Red Hat Linux CD Part 1 into the CD-ROM drive

```
[root@deep /]# mount /dev/cdrom /mnt/cdrom
[root@deep /]# cd /mnt/cdrom/images/
[root@deep images]# dd if=boot.img of=/dev/fd0H1440 bs=1440k
1+0 records in
1+0 records out
[root@deep images]# cd /
[root@deep /]# umount /mnt/cdrom
```

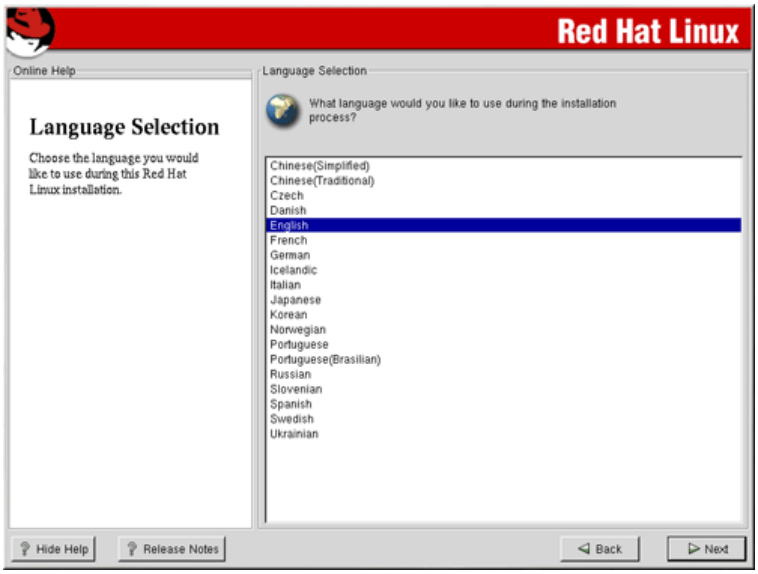
Don’t forget to label the diskette “Red Hat boot disk”, for example.

Beginning the installation of Linux

Now that we have made the boot disk, it is time to begin the installation of Linux. Since we'd start the installation directly off the CD-ROM, boot with the boot disk. Insert the boot diskette you create into the drive A: on the computer where you want to install Linux and reboot the computer. At the `boot:` prompt, press **`Enter`** to continue booting and follow the three simple steps below.

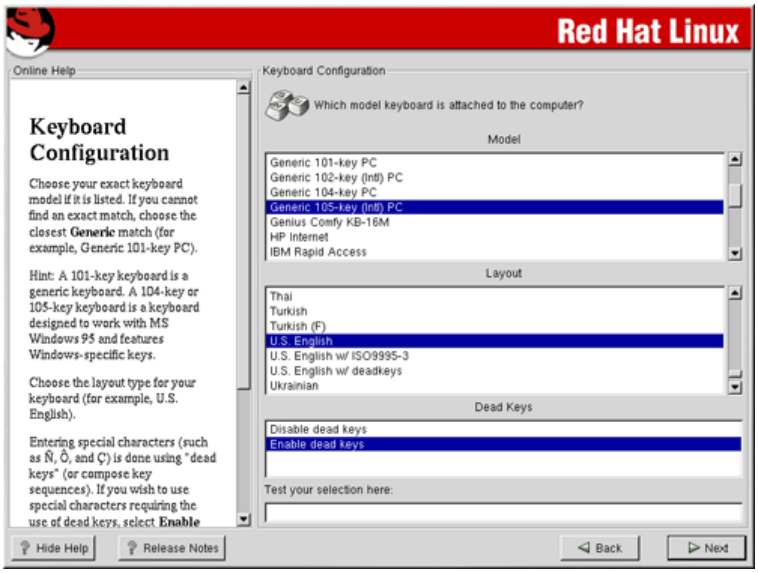
Step 1

The first step is to choose what language should be used during the installation process. In our example we choose the English language. Once you select the appropriate language, click **Next** to continue.

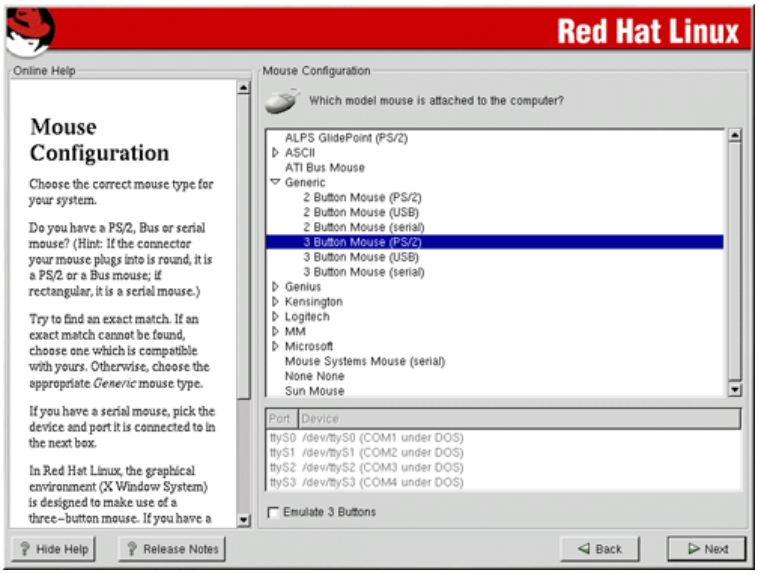


Step 2

Next, the system allows you to choose your keyboard type, layout type for the keyboard, and the possibility to enable or disable Dead Keys. Once you have made the appropriate selections, click **Next** to continue.



Step 3
Finally, we choose the kind of mouse type we have and if this mouse has two or three buttons. If you have a mouse with just two buttons, you can select the option named "**Emulate 3 Buttons**" and click both mouse buttons at the same time to act as the middle mouse button.



Once we have completed the above three steps, we are ready to begin the installation of Red Hat Linux.

Installation Class and Method (Install Options)

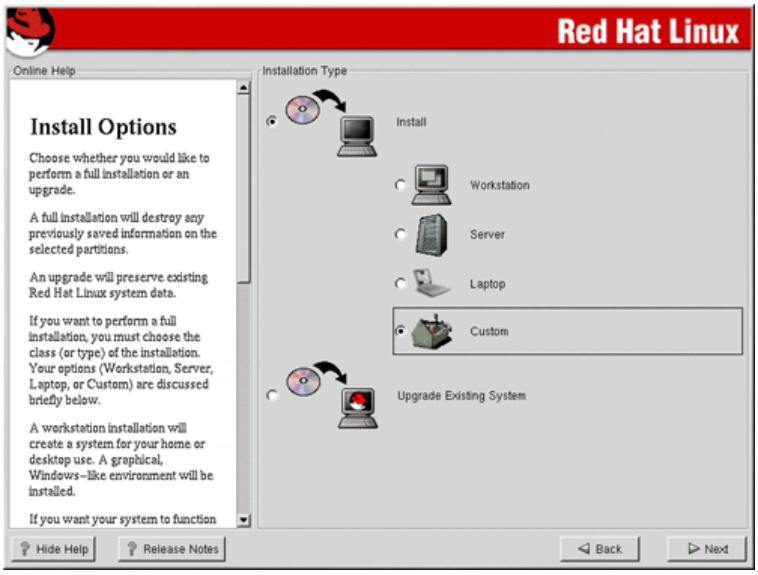
Red Hat Linux 7.3 includes four different classes, or type of installation. They are:

- ✓ Workstation
- ✓ Server
- ✓ Laptop
- ✓ Custom

The first two classes (Workstation and Server) give you the option of simplifying the installation process with a significant loss of configuration flexibility that we don't want to lose.

For this reason we highly recommend you select the "**Custom**" installation. Only the custom-class installation gives us complete flexibility. During the custom-class installation, it is up to you how disk space should be partitioned. We also have complete control over the different `RPM` packages that will be installed on the system.

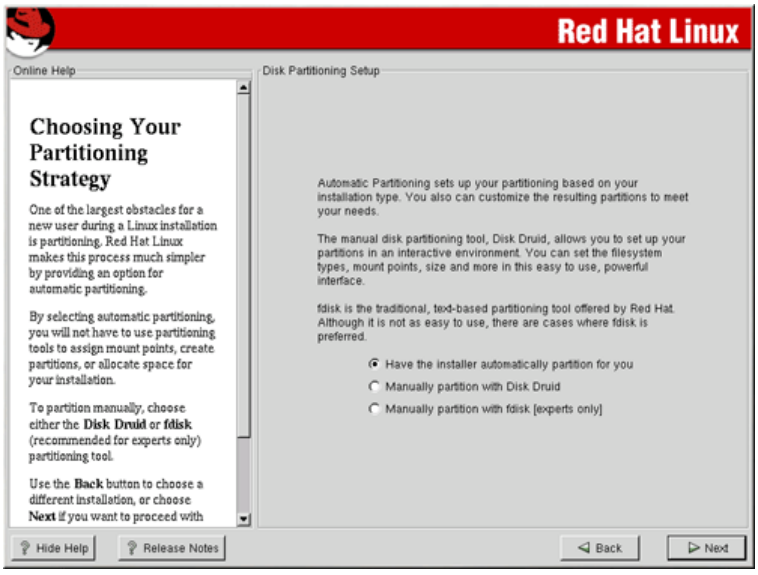
The idea is to load the minimum amount of packages, while maintaining maximum efficiency. The less software that resides on the machine, the fewer potential security exploits or holes may appear. From the menu that appears on your screen, select the "**Custom**" installation class and click **Next**.



Partition your system for Linux

Partitioning allows you to divide your hard drive into isolated sections, where each section behaves as its own hard drive. This is a useful security measure and to avoid some possible DoS attacks because we can create separate partition for specific services that we would like to run on our Linux server. See later in this book for more information about which partition strategy to use with security.

The system will show you a new screen from where you can choose the tool you would like to use to partition the disks for Linux.



From here we have two choices, but before we explain them, it is important to understand partition strategies first.

We assume that you are installing the new Linux server to a new hard drive, with no other existing file system or operating system installed. A good partition strategy is to create a separate partition for each major file system. This enhances security and prevents accidental **D**enial **o**f **S**ervice (DoS) or exploit of SUID programs.

Creating multiple partitions offers you the following advantages:

- ✓ Protection against **D**enial **o**f **S**ervice attack.
- ✓ Protection against SUID programs.
- ✓ Faster booting.
- ✓ Easy backup and upgrade management.
- ✓ Ability for better control of mounted file system.
- ✓ Limit each file system's ability to grow.
- ✓ Improve performance of some program with special setup.

WARNING: If a previous file system or operating system exists on the hard drive and computer where you want to install your Linux system, we highly recommend, that you make a backup of your current system before proceeding with the disk partitioning.

Partitions Strategy

For performance, stability and security reasons you must create something like the following partitions listed below on your computer. We suppose for this partition configuration the fact that you have a SCSI hard drive of 9.1 GB with 256 MB of physical RAM. Of course you will need to adjust the partition sizes and swap space according to your own needs and disk size.

Minimal recommended partitions that must be created on your system:

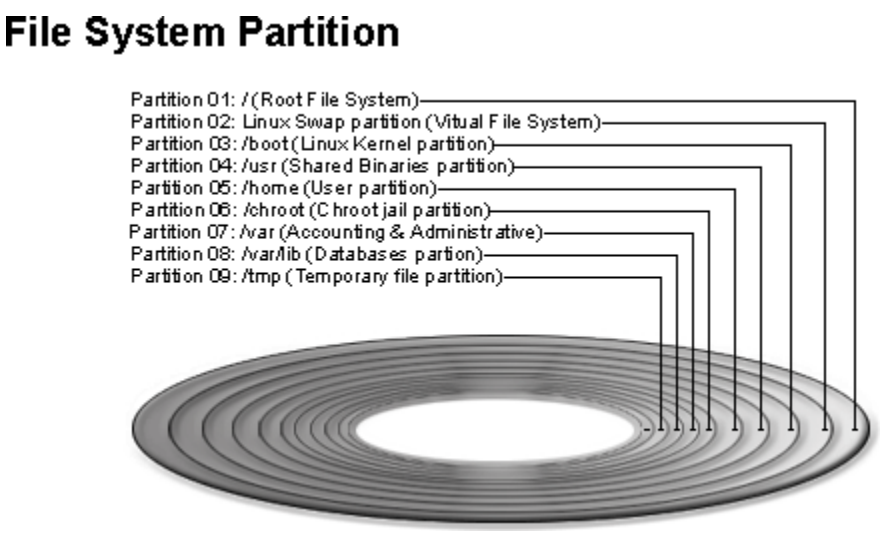
This is the minimum number of partitions we recommend creating whatever you want to setup it for, a Web Server, Mail Server, Gateway or something else.

/boot	5	MB	All Kernel images are kept here.
/	256	MB	Our root partition.
/usr	512	MB	Must be large, since many Linux binaries programs are installed here.
/home	5700	MB	Proportional to the number of users you intend to host. (i.e. 100 MB per users * by the number of users 57 = 5700 MB)
/var	256	MB	Contains files that change when the system run normally (i.e. Log files).
/tmp	329	MB	Our temporary files partition (must always reside on its own partition).
<Swap>	512	MB	Our swap partition. The virtual memory of the Linux operating system.

Additional or optional partitions that can be created on your system:

Depending on what services the Linux system will be assigned to serve or the specific software requirements, there can be some special partitions you can add to the minimum partitions we recommend. You can create as many partitions as you want to fit you needs. What we show you below are partitions related to programs we describe in the book.

/chroot	256	MB	If you want to install programs in chroot jail environment (i.e. DNS, Apache).
/var/lib	1000	MB	Partition to handle SQL or Proxy Database Server files (i.e. MySQL, Squid).



All major file systems are on separate partitions

As you can see, there are two partitions, which are less common than the others. Let's explain each of them in more detail:

The `/chroot` partition can be used for `DNS` Server chrooted, `Apache` web server chrooted and other chrooted future programs. The `chroot()` command is a Unix system call that is often used to provide an additional layer of security when untrusted programs are run. The kernel on Unix variants which support `chroot()` maintains a note of the root directory each process on the system has. Generally this is `/`, but the `chroot()` system call can change this. When `chroot()` is successfully called, the calling process has its idea of the root directory changed to the directory given as the argument to `chroot()`.

The `/var/lib` partition can be used to handle `SQL` or `Squid` Proxy database files on the Linux server. This partition can be useful to limit accidental **D**enial **o**f **S**ervice attack and to improve the performance of the program by tuning the `/var/lib` file system.

Putting `/tmp` and `/home` on separate partitions is pretty much mandatory if users have shell access to the server (protection against `SUID` programs), splitting these off into separate partitions also prevents users from filling up critical file systems (denial of service attack), putting `/var`, and `/usr` on separate partitions is also a very good idea. By isolating the `/var` partition, you protect your root partition from overfilling (**D**enial **o**f **S**ervice attack).

In our partition configuration we'll reserve 256 MB of disk space for chrooted programs like `Apache`, `DNS` and other software. This is necessary because `Apache DocumentRoot` files and other binaries, programs related to it will be installed in this partition if you decide to run `Apache` web server in a chrooted jail. Note that the size of the `Apache` chrooted directory on the chrooted partition is proportional to the size of your `DocumentRoot` files or number of users.

NOTE: It is for you to decide how much disk space should be reserved and set for each partition you may need to create on your server. The choice completely depends on you and your computer hardware. If you have a lot of disk space and know that you will need to run many services in chroot jail environment, then you can decide to reserve more space for the chroot jail structure on your system.

Swap related issues:

Swap relates to virtual RAM on the system. This special device is needed when you run out of physical RAM because you don't have enough MB of RAM available or your applications required more than what is available on your computer. It is not true that swap space is needed on every system, but to ensure that you do not run out of swap, it is recommended to create a swap partition on the server.

The 2.4 kernel of Linux is more aggressive than the 2.2 kernels in its use of swap space and the optimal sizing of swap space remains dependent on the following:

1. The amount of RAM installed.
2. The amount of disk space available for swap.
3. The applications being run.
4. The mix of applications that are run concurrently.

No rule-of-thumb can possibly take all these points into account. However, we recommend the following swap sizes:

- Single-user systems with less than 128MB physical RAM: 256MB
- Single-user systems and low-end servers with more than 128MB physical RAM: two times physical RAM (2xRAM)
- Dedicated servers with more than 512MB physical RAM: highly dependent on environment and must be determined on a case-by-case basis)

NOTE: Swap is bad and it is recommended that you try to avoid it as much as possible by installing more physical RAM whenever possible. If you see that your system begin to swap memory, then consider buying some more RAM. Remember that swap is bad and your rules are to avoid it as much as possible for optimum performance of your Linux server.

Minimum size of partitions for very old hard disk:

For information purposes only, this is the minimum size in megabytes, which a Linux installation must have to function properly. The sizes of partitions listed below are really small. This configuration can fit into a very old hard disk of 512MB in size that you might find in old i486 computers. We show you this partition just to get an idea of the minimum requirements.

```
/          35MB
/boot      5MB
/chroot    10MB
/home      100MB
/tmp       30MB
/usr       232MB
/var       25MB
```

WARNING: Trying to compile programs on a 512 MB hard drive, will fail due to the lack of available space. Instead, install RPM's packages.

Disk Partition (Manual Partitioning)

Now that we know exactly what partitions we need to create for our new Linux server, it is time to choose the partitioning software we will use to make these partitions. With Red Hat Linux two programs exist to assist you with this step:

- Manually partition with Disk druid
- Manually partition with fdisk [experts only]

`Disk Druid` is new software used by default in Red Hat Linux to partition your disk drive, this program is easy to use, and allows you to use a graphical interface to create your partitions tables.

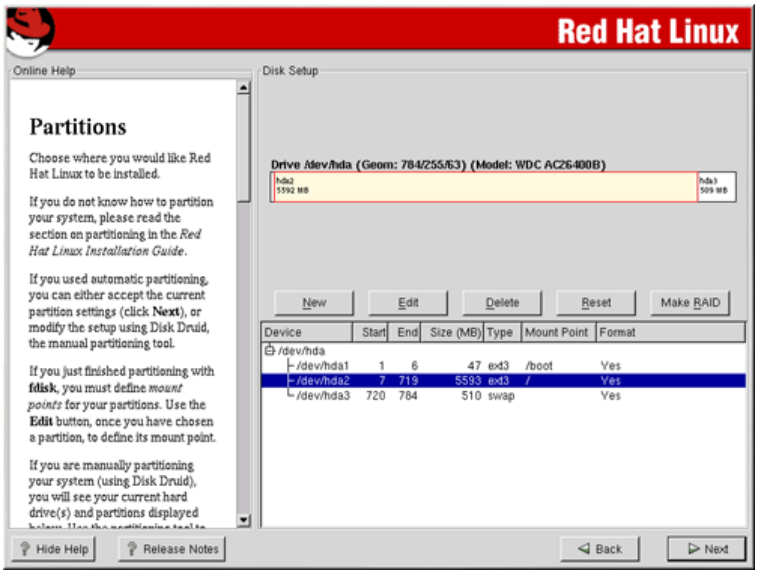
`fdisk` was the first partitioning program available on Linux. It is more powerful then `Disk Druid` and allows you to create your partition table in exactly the way you want it (if you want to put your swap partition near the beginning of your drive, then you will need to use `fdisk`). Unfortunately, it is also a little more complicated than `Disk Druid` and many Linux users prefer to use `Disk Druid` for this reason.

Personally, I prefer to create the partitions with the `fdisk` program and I recommend you use and be familiar with it, because if, in the future you want to add or change some file systems you will need to use `fdisk`.

Partitioning with `Disk Druid`

This section applies only if you chose to use **`Disk Druid`** to partition your system. `Disk Druid` is a program that partitions your hard drive for you. Choose "**`New`**" to add a new partition, "**`Edit`**" to edit a partition, "**`Delete`**" to delete a partition and "**`Reset`**" to reset the partitions to the original state. When you add a new partition, a new window appears on your screen and gives you parameters to choose.

Mount Point: for where you want to mount your new partition in the filesystem.
Filesystem Type: `Ext3` for Linux filesystem and `Swap` for Linux `Swap` Partition
Size (MB): for the size of your new partition in megabytes.



If you have a SCSI disk, the device name will be /dev/sda and if you have an IDE disk it will be /dev/hda. If you're looking for high performance and stability, a SCSI disk is highly recommended.

Linux refers to disk partitions using a combination of letters and numbers. It uses a naming scheme that is more flexible and conveys more information than the approach used by other operating systems.

Here is a summary:

First Two Letters – The first two letters of the partition name indicate the type of device on which the partition resides. You'll normally see either **hd** (for IDE disks), or **sd** (for SCSI disks).

The Next Letter – This letter indicates which device the partition is on. For example: /dev/hda (the first IDE hard disk) and /dev/hdb (the second IDE disk), etc.

Keep this information in mind, it will make things easier to understand when you're setting up the partitions Linux requires.

Now, as an example:

To make the partitions listed below on your system (this is the partition we'll need for our server installation example); the commands below are for Disk Druid:

Step 1
Execute all of the following commands with Disk Druid to create the require partitions.

```
New
Mount Point: /boot
Filesystem Type: ext3
Size (Megs): 24
Ok

New
Mount Point: /
Filesystem Type: ext3
Size (Megs): 256
Ok

New
Mount Point: /usr
Filesystem Type: ext3
Size (Megs): 512
Ok

New
Mount Point: /home
Filesystem Type: ext3
Size (Megs): 4512
Ok

New
Mount Point: /chroot
Filesystem Type: ext3
Size (Megs): 256
Ok
```

```
New
Mount Point: /var
Filesystem Type: ext3
Size (Megs): 512
Ok

New
Mount Point: /var/lib
Filesystem Type: ext3
Size (Megs): 1024
Ok

New
Mount Point: /tmp
Filesystem Type: ext3
Size (Megs): 256
Ok

New
Mount Point: swap
Filesystem Type: swap
Size (Megs): 1372
Ok
```

Step 2
After you have executed the above commands to create and partition your drive with **`Disk Druid`**, press the **Next** button and continue the installation to choose partitions to format.

Partitioning with `fdisk`

This section applies only if you chose to use **`fdisk`** to partition your system.

The first thing you will want to do is using the **`p`** key to check the current partition information. You need to first add your root partition. Use the **`n`** key to create a new partition and then select either **`e`** or **`p`** keys for extended or primary partition.

Most likely you will want to create a primary partition. You are asked what partition number should be assigned to it, at which cylinder the partition should start (you will be given a range – **just choose the lowest number (1)**), and the size of the partition. For example, for a 5MB partition, you would enter +5M for the size when asked.

Next, you need to add your extended partition. Use the **`n`** key to create a new partition and then select the **`e`** key for extended partition. You are asked what partition number should be assigned to it, at which cylinder the partition should start (you will be given a range – **just choose the lowest number (2)**), and the size of the partition. **You would enter the last number for the size when asked (or just press Enter)**.

You will now want to create the swap partition. You need to use the **`n`** key for a new partition. Choose logical; tell it where the first cylinder should be **(2)**. Tell `fdisk` how big you want your swap partition. You then need to change the partition type to `Linux swap`. Enter the **`t`** key to change the type and enter the partition number of your swap partition. Enter the number **`82`** for the hex code for the `Linux swap` partition.

Now that you have created your Linux boot and Linux swap partition, it is time to add any additional partitions you might need. Use the **n** key again to create a new partition, and enter all the information just as before. Keep repeating this procedure until all your partitions are created. You can create up to four primary partitions; then you must start putting extended partitions into each primary partition.

NOTE: None of the changes you make take effect until you save then and exit `fdisk` using the **w** command. You may quit `fdisk` at any time without saving changes by using the **q** command.

An overview of `fdisk`

- The command for help is **m**
- To list the current partition table, use **p**
- To add a new partition, use **n**
- To delete a partition, use **d**
- To set or changes the partition type, use **t**
- To provide a listing of the different partition types and their ID numbers, use **l**
- To saves your information and quits `fdisk`, use **w**

Now, as an example:

To make the partitions listed below on your system (these are the partitions we'll need for our server installation example); the commands below are for `fdisk`:

Step 1
Execute all of the following commands with `fdisk` to create the require partitions.

```
Command (m for help): n
Command action
    e extended
    p primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-1116, default 1): 1
Last cylinder or +size or +sizeM or +sizeK (1-1116, default 1116): +18M

Command (m for help): n
Command action
    e extended
    p primary partition (1-4)
e
Partition number (1-4): 2
First cylinder (4-1116, default 4): 4
Last cylinder or +size or +sizeM or +sizeK (4-1116, default 1116): 1116

Command (m for help): n
Command action
    l logical (5 or over)
    p primary partition (1-4)
l
First cylinder (4-1116, default 4): 4
Last cylinder or +size or +sizeM or +sizeK (4-1116, default 1116): +256M
```

```
Command (m for help): n
Command action
    l logical (5 or over)
    p primary partition (1-4)
l
First cylinder (37-1116, default 37): 37
Last cylinder or +size or +sizeM or +sizeK (37-1116, default 1116): +512M

Command (m for help): n
Command action
    l logical (5 or over)
    p primary partition (1-4)
l
First cylinder (103-1116, default 103): 103
Last cylinder or +size or +sizeM or +sizeK (103-1116, default 1116): +4512M

Command (m for help): n
Command action
    l logical (5 or over)
    p primary partition (1-4)
l
First cylinder (679-1116, default 679): 679
Last cylinder or +size or +sizeM or +sizeK (679-1116, default 1116): +256M

Command (m for help): n
Command action
    l logical (5 or over)
    p primary partition (1-4)
l
First cylinder (712-1116, default 712): 712
Last cylinder or +size or +sizeM or +sizeK (712-1116, default 1116): +512M

Command (m for help): n
Command action
    l logical (5 or over)
    p primary partition (1-4)
l
First cylinder (778-1116, default 778): 778
Last cylinder or +size or +sizeM or +sizeK (778-1116, default 1116): +1024M

Command (m for help): n
Command action
    l logical (5 or over)
    p primary partition (1-4)
l
First cylinder (909-1116, default 909): 909
Last cylinder or +size or +sizeM or +sizeK (909-1116, default 1116): +256M

Command (m for help): n
Command action
    l logical (5 or over)
    p primary partition (1-4)
l
First cylinder (942-1116, default 942): 942
Last cylinder or +size or +sizeM or +sizeK (942-1116, default 1116): 1116

Command (m for help): t
Partition number (1-12): 12
Hex code (type L to list codes): 82
Changed system type of partition 12 to 82 (Linux swap)
```

Step 2
Now, use the **p** command to list the partition that we've created, you must see something like the following information on your screen.

```
Command (m for help): p

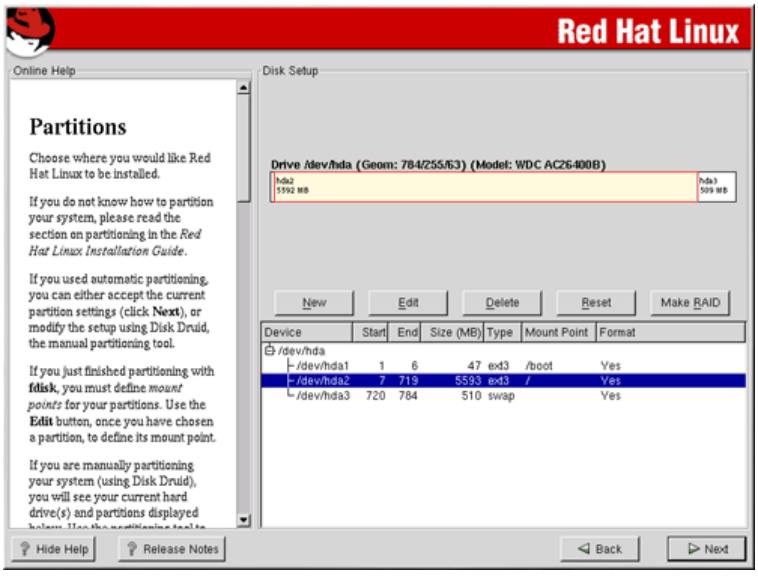
Disk /tmp/sda: 255 heads, 63 sectors, 1116 cylinders
Units = cylinders of 16065 * 512 bytes

Device Boot   Start         End          Blocks       Id           System
/tmp/sda1     1           3            24066        83           Linux
/tmp/sda2     4           1116         8940172+     5            Extended
/tmp/sda5     4           36           265041       83           Linux
/tmp/sda6     37          102          530113+      83           Linux
/tmp/sda7     103         678          4626688+     83           Linux
/tmp/sda8     679         711          265041       83           Linux
/tmp/sda9     712         777          530113+      83           Linux
/tmp/sda10    778         908          1052226      83           Linux
/tmp/sda11    909         941          265041       83           Linux
/tmp/sda12    942         1116         1405656      82           Linux Swap
```

Step 3
If all the partitions look fine and meet your requirements, use the **w** command to write the table to disk and exit `fdisk` program:

```
Command (m for help): w
The partition table has been altered
```

Step 4
After you have partitioned your drive with **fdisk**, press **Next** and continue the installation with **Disk Druid** to choose the mount point of the directories. `Disk Druid` contains a list of all disk partitions with file-systems readable by Linux. This gives you the opportunity to assign these partitions to different parts of your Linux system when it boots. Select the partition you wish to assign and press **Enter**; then enter the mount point for that partition, e.g., `/var`.



Boot Loader Installation

On the next screen you will see the Boot Loader Configuration screen. In order to boot your Linux system, you usually need to install a boot loader. With new release of Linux, you can choose to install either GRUB, LILO, or you can choose not to install a boot loader at all.

GRUB is the new and recommended method to boot Linux. You can still decide to use LILO, but it's better to go with GRUB now. From this screen, you will see different configurable options related to GRUB or LILO.

The first option is:

- Use GRUB as the boot loader

This option allows you to use the GRUB software as your boot loader to boot your Linux operating system on the computer. This is the recommended method to use with Linux. GRUB works in the same way as LILO work with many additional security and advanced features that LILO cannot provide you. In our setup, we use this option to boot our Linux server.

The second option is:

- Use LILO as the boot loader

This option allows you to use the LILO software as your boot loader to boot your Linux operating system on the computer. Remember that LILO is now the old method to boot Linux and I recommend you to go with GRUB instead if you want to stay up-to-date with latest technology on the Linux world. In our setup, we don't choose or use this option.

The third option is:

- Do not install a boot loader

This option allows you to skip installing any type of available boot loader (GRUB or LILO) with Linux. This is useful if you use a boot disk rather than GRUB or LILO to start your operating system. This can greatly improve security in some case since you need to have a bootable Linux floppy with the kernel on it to start the server. But in other hand, you will not be able to restart the server remotely if something happens. In our setup, we don't use this option.

The fourth option is:

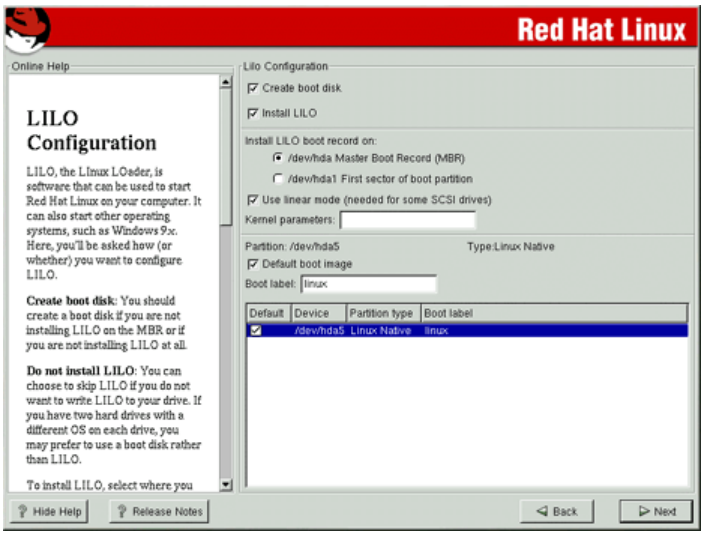
- Install Boot Loader record on:
 - ✓ **M**aster **B**oot **R**ecord (MBR)
 - ✓ First sector of boot partition

Usually, if Linux is the only operating system on your machine (and this must be the case in a server installation), you should choose the "**Master Boot Record (MBR)**" option. The MBR is a special area on your hard drive that is automatically loaded by your computer's BIOS, and is the earliest point at which the boot loader can take control of the boot process.

The fifth option is:

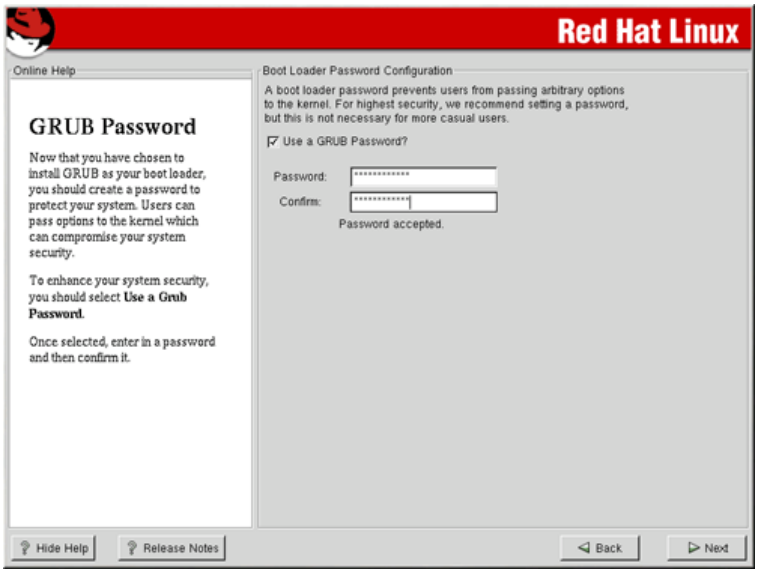
- Force use of `LBA32`

This option (if checked) allows you to exceed the 1024 cylinder limit for the `/boot` partition. If you have a system which supports the `LBA32` extension for booting operating systems above the 1024 cylinder limit, and you want to place your `/boot` partition above cylinder 1024, you should select this option but in most case you can live without it and your system will perfectly work. In our setup of the operating system, we don't use it.



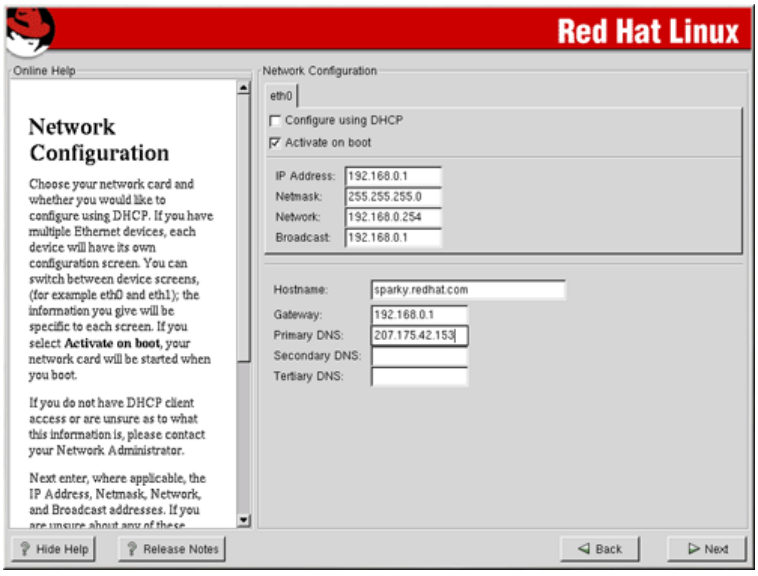
The `GRUB` Password

This section applies only if you have selected `GRUB` as your boot loader. If you are installing `GRUB` as your boot loader, you should create a password to protect your system. Without a `GRUB` password, users with access to your system can pass options to the kernel which can compromise your system security. With a `GRUB` password in place, the password must first be entered in order to select any non-standard boot options.



Network Configuration

After that, you need to configure your network. If you have multiple Ethernet devices, each device will have its own configuration screen. You will be answered to enter the `IP` Address, Netmask, Network, Broadcast addresses, and the Gateway, Primary `DNS` (and if applicable the Secondary `DNS` and Ternary `DNS`) addresses. You should know all of the information or you can ask your system administrator to help you get the correct information.

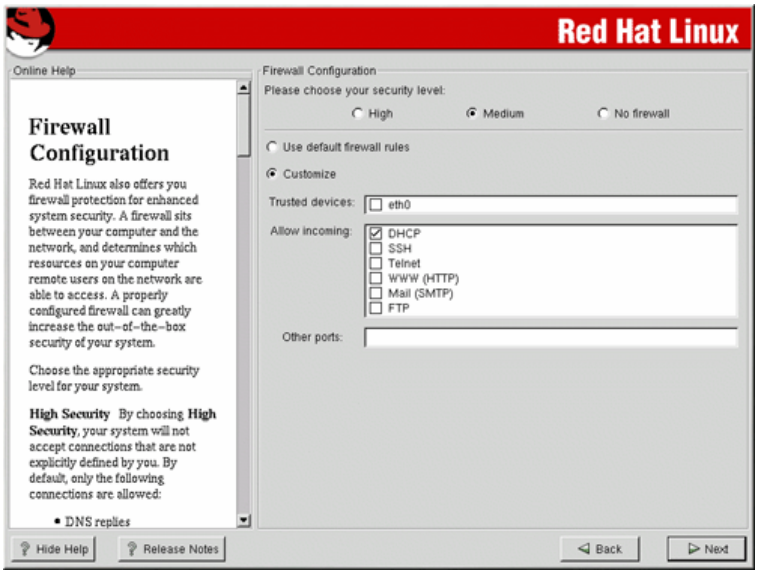


Firewall Configuration

From this part of the setup installation, we have possibility to configure a Firewall. This is OK for the average end user but **NOT** for serious Firewall security. This newly added feature uses the old `IPCHAINS` tool of Linux with the help of a small utility named "`lokkit`" to set up your firewall.

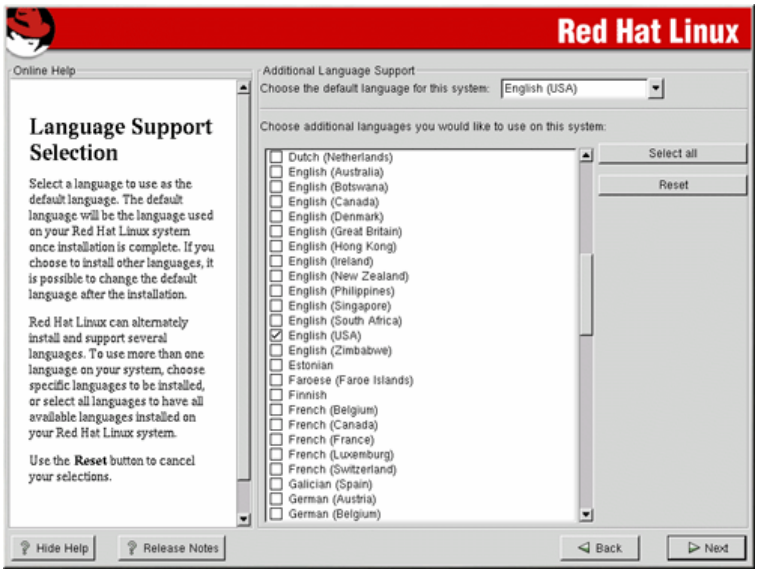
I highly recommend you to deactivate this feature now and see later in this book on how to install and configure `IPTables` with `GIPTable`, which is the new Firewall tool to use with Linux and kernel 2.4 generation. `GIPTables` is simply a Firewall software that can help you to configure `IPTables` in the most secure and easily way than any other firewall software can provide you.

From the next screen that appears, you will see three different security levels available, choose the "**No firewall**" option and click **Next**.



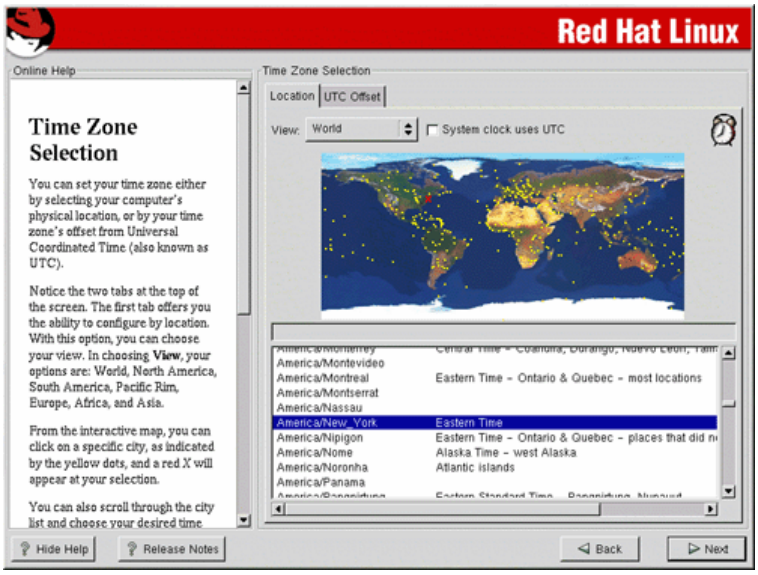
Language Support Selection

With the internalization, a need for different language support has appeared. From here the installation will ask you to choose the default language that will be used on your Linux system once the installation is complete. If you are only going to use one language on your system, selecting only this language will save significant disk space.



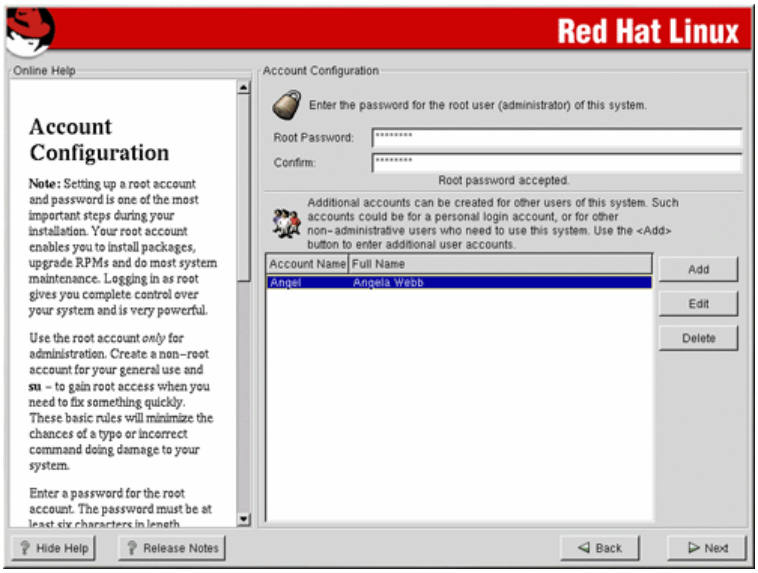
Time Zone Selection

On the next screen, you will have the opportunity to set your time zone. Once selected click **Next**.



Account Configuration

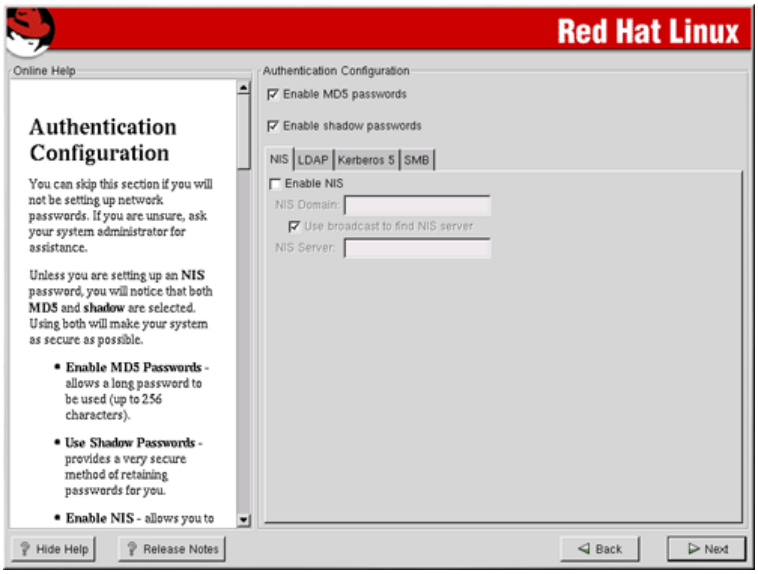
After the clock has been configured, you need to give your system a root password account.



Authentication Configuration

Finally, the last stage is the authentication configuration. For Authentication Configuration don't forget to select:

- ✓ Enable MD5 passwords
- ✓ Enable Shadow passwords



Enable MD5 passwords - allows a long password to be used (up to 256 characters), instead of the Unix standard eight letters or less.

Enable shadow passwords - provides a very secure method of retaining passwords for you. All passwords are stored in a file named shadow, which is readable only by the super-user root.

Enable NIS, LDAP, Kerberos and SMB doesn't need to be selected since we are not configuring these services on this server right know.

Selecting Package Groups

After your partitions have been configured and selected for formatting and configurations have been set for your specific system, you are ready to select packages for installation. By default, Linux is a powerful operating system that runs many useful services. However, many of these services are unneeded and pose potential security risks.

Ideally, each network service should be on a dedicated, single-purpose host. Many Linux operating systems are configured by default to provide a wider set of services and applications than are required to provide a particular network service, so you may need to configure the server to eliminate unneeded services. Offering only essential services on a particular host can enhance your network security in several ways:

- ✓ Other services cannot be used to attack the host and impair or remove desired network services.

- ✓ The host can be configured to better suit the requirements of the particular service. Different services might require different hardware and software configurations, which could lead to needless vulnerabilities or service restrictions.

- ✓ By reducing services, the number of logs and log entries is reduced so detecting unexpected behavior becomes easier.

- ✓ Different individuals may administer different services. By isolating services so each host and service has a single administrator you will minimize the possibility of conflicts between administrators.

A proper installation of your Linux server is the first step to a stable, secure system. From the screen menu that appears (Selecting Package Groups), you first have to choose which system components you want to install, in our case; we must **DESELECT ALL CHECKED** Package Groups on the list.

Since we are configuring a Linux server, we don't need to install a graphical interface (`XFree86`) on our system (a graphical interface on a server means less processes, less `CPU` availability, less memory, security risks, and so on), also computers are subject to the treachery of images as well. The image on your computer screen is not a computer file -- it's only an image on a computer screen. Images of files, processes, and network connections are very distant cousins of the actual bits in memory, in network packets, or on disks.

Layer upon layer of hardware and software produces the images that you see. When an intruder "owns" a machine, any of those layers could be tampered with. Application software can lie, OS kernels can lie, boot PROMs can lie, and even hard disk drives can lie. Graphical interfaces are usually used on only workstations.

Step 1
First of all, it is vital to verify and be **SURE** to deselect all of the following Package Group:

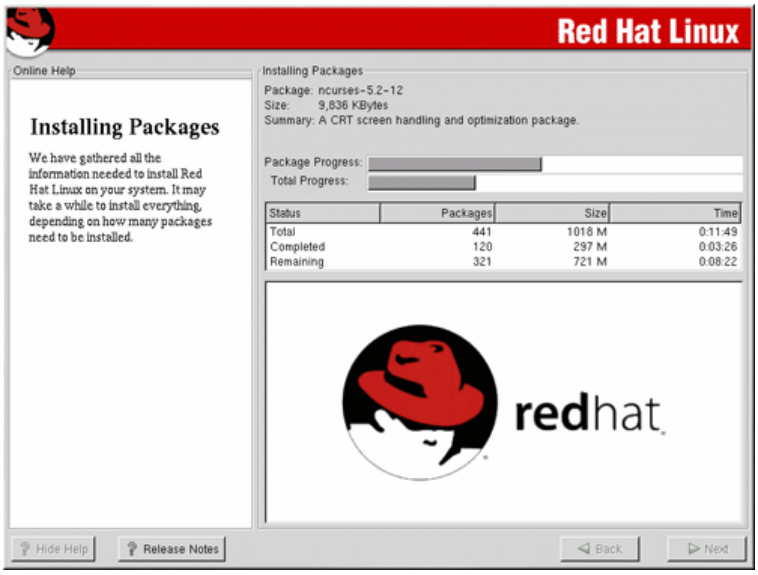
- ✓ Printing Support
- ✓ Classic X Window System
- ✓ X Window System
- ✓ Laptop Support
- ✓ GNOME
- ✓ KDE
- ✓ Sound and Multimedia Support
- ✓ Network Support
- ✓ Dialup Support
- ✓ Messaging and Web Tools
- ✓ Graphics and Image Manipulation
- ✓ New Server
- ✓ NFS File Server
- ✓ Windows File Server
- ✓ Anonymous FTP Server
- ✓ SQL Database Server
- ✓ Web Server
- ✓ Router / Firewall
- ✓ DNS Name Server
- ✓ Network Managed Workstation
- ✓ Authoring and Publishing
- ✓ Emacs
- ✓ Utilities
- ✓ Legacy Application Support
- ✓ Software Development
- ✓ Kernel Development
- ✓ Windows Compatibility / Interoperability
- ✓ Games and Entertainment
- ✓ Everything

To resume, it is very important and I say VERY IMPORTANT to deselect (none is selected) every selected Packages Group before clicking on the **Next** button for continuing the installation.

We don’t want and don’t need to install any additional packages. The default install of this Linux distribution already comes with the most essential programs we need for the base functionality of the operating system.



Step 2
At this point, the installation program will check dependencies in packages selected for installation (in our case no packages are selected) and format every partition you selected for formatting in you system. This can take several minutes depending on the speed of your machine. Once all partitions have been formatted, the installation program starts to install Linux to your hard drive.



Boot Disk Creation

From this section of the installation, we have the possibility to create a boot disk for our newly installed operating system. If you do not want to create a boot disk, you should check the "**Skip boot disk creation**" checkbox before you click **Next**. Creating a boot disk must be made if you decide to not install `GRUB` or `LILO` on the `MBR` (the **M**aster **B**oot **R**ecord) or if you are not installing `GRUB` or `LILO` at all.



How to use `RPM` Commands

This section contains an overview of using `RPM` for installing, uninstalling, upgrading, querying, listing, and checking `RPM` packages on your Linux system. You must be familiar with these `RPM` commands now because we'll use them often in this book and especially later in this chapter for software that must be uninstalled after installation of the server.

Install a `RPM` package:

Note that `RPM` packages have a file of names like **`foo-1.0-2.i386.rpm`**, which include the package name (**`foo`**), version (**`1.0`**), release (**`2`**), and architecture (**`i386`**).

- To install a `RPM` package, use the command:

```
[root@deep /]# rpm -ivh foo-1.0-2.i386.rpm
foo                  ##################################################
```

Uninstall a `RPM` package:

Notice that we used the package name "**`foo`**", not the name of the original package file "**`foo-1.0-2.i386.rpm`**".

- To uninstall a `RPM` package, use the command:

```
[root@deep /]# rpm -e foo
```

Upgrade a RPM package:
With this command, RPM automatically uninstalls the old version of foo package and installs the new one. Always use "rpm -Uvh" command to install packages, since it works fine even when there are no previous versions of the package installed. This is the recommended method of installing package on the system.

- To upgrade a RPM package, use the command:

```
[root@deep /]# rpm -Uvh foo-1.0-2.i386.rpm
foo                  ##################################################
```

Force the installation of a RPM package:
With this command, RPM will force the installation of the specified package even if some conflict or other kind of problem exists. This command should be used with care and only if you know what you do. In most case, RPM can correctly guest problem and refuse to install. To bypass RPM warning, you can use the RPM command below.

- To force the installation of a RPM package, use the command:

```
[root@deep /]# rpm -Uvh --force foo-1.0-2.i386.rpm
foo                  ##################################################
```

Avoid RPM package dependency:
With this command, RPM will not take care of package dependency and will install the RPM software on your system. Package dependency is an important concept in the RPM world. Dependency is when some other packages depend of the RPM package you are trying to install.

By default, RPM check if all other RPM packages required for the RPM you try to install are present before installing the RPM. If some required packages are not present, RPM will inform you. This is made to avoid problem and be sure that the software you want to install will perfectly work. In some special case, we don't need to take care of dependency and can use the option below to inform it to skip the dependency check when installing the software.

- To avoid RPM package dependency, use the command:

```
[root@deep /]# rpm -Uvh --nodeps foo-1.0-2.i386.rpm
foo                  ##################################################
```

Query a RPM package:
This command will print the package name, version, and release number of installed package foo. Use this command to verify that a package is or is not installed on your system.

- To query a RPM package, use the command:

```
[root@deep /]# rpm -q foo
foo-2.3-8
```

Display RPM package information:
This command displays package information; includes name, version, and description of the installed program. Use this command to get information about the installed package.

- To display RPM package information, use the command:

```
[root@deep /]# rpm -qi foo
Name        : foo                Relocations: none
Version     : 2.3                Vendor: OpenNA.com, Inc.
Release     : 8                  Build Date: Thu 24 Aug 2000 11:16:53 AM EDT
Install date: Mon 12 Feb 2001 01:17:24 AM EST      Build Host: openna.com
Group       : Applications/Archiving        Source RPM: foo-2.3-8.src.rpm
Size        : 271467                           License: distributable
Packager    : OpenNA.com, Inc. <http://www.openna.com/>
Summary     : Here will appears summary of the package.
Description : Here will appears the description of the package.
```

Display RPM package information before installing the program:
This command displays package information; includes name, version, and description of the program without the need to install the program first. Use this command to get information about a package before you install it on your system.

- To display package information before installing the program, use the command:

```
[root@deep /]# rpm -qpi foo-2.3-8.i386.rpm
Name        : foo                Relocations: none
Version     : 2.3                Vendor: OpenNA.com, Inc.
Release     : 8                  Build Date: Thu 24 Aug 2000 11:16:53 AM EDT
Install date: Mon 12 Feb 2001 01:17:24 AM EST      Build Host: openna.com
Group       : Applications/Archiving        Source RPM: foo-2.3-8.src.rpm
Size        : 271467                           License: distributable
Packager    : OpenNA.com, Inc. <http://www.openna.com/>
Summary     : Here will appears summary of the package.
Description : Here will appears the description of the package.
```

List files in a installed RPM package:
This command will list all files in a installed RPM package. It works only when the package is already installed on your system.

- To list files in a installed RPM package, use the command:

```
[root@deep /]# rpm -ql foo
/usr/bin/foo
/usr/bin/foo1
/usr/sbin/foo2
```

List files in RPM package that is not already installed:
This command will list all files in a RPM package that is not already installed on your system. It is useful when you want to know which components are included in the package before installing it.

- To list files in RPM package that is not already installed, use the command:

```
[root@deep /]# rpm -qpl foo
/usr/lib/foo
/usr/bin/foo1
/usr/sbin/foo2
```

Know which files is part of which RPM package:
This command will show you from which RPM package the file comes from. It works only when the package is already installed on your system and it is very useful when you see some files into Linux that you do not know about it and want to get more information about its RPM provenance.

- To know which files is part of which RPM package, use the command:
```
[root@deep /]# rpm -qf /etc/passwd
setup-2.3.4-1
```

Check RPM signature package:
This command checks the PGP signature of specified package to ensure its integrity and origin. Always use this command first before installing new RPM package on your system. GnuPG or PGP software must be already installed on your system before you can use this command. See the chapter related to GnuPG installation and configuration for more information.

- To check a RPM signature package, use the command:
```
[root@deep /]# rpm --checksig foo
```

Examine the md5sum of RPM package:
The RPM md5sum is useful to verify that a package has not been corrupted or tampered with. You can use it to be sure that the download of your new RPM package was not corrupted during network transfer.

- To examine only the md5sum of the package, use the command:
```
[root@deep /]# rpm --checksig --nogpg foo
```

Starting and stopping daemon services
The **init** program of Linux (also known as process control initialization) is in charge of starting all the normal and authorized processes that need to run at boot time on your system. These may include the APACHE daemons, NETWORK daemons, and anything else that must be running when your machine boots.

Each of these processes has a script file under the /etc/init.d directory written to accept an argument, which can be start, stop, restart, etc. As you can imagine, those script are made to simplify the administration of the server and the way we can start or stop services under Linux. Of course, we can use the native way to start all required services under our server, but it is much simple to have some kind of script files that should provide us some easy method to automate and control the procedures. This is why **init** program and all initialization script files available under the /etc/init.d directory exist.

Below are some examples showing you how to execute those scripts by hand.

For example:

- To start the httpd web server daemon manually under Linux, you'll type:
```
[root@deep /]# /etc/init.d/httpd start
Starting httpd:              [OK]
```

- To stop the httpd web server daemon manually under Linux, you'll type:
```
[root@deep /]# /etc/init.d/httpd stop
Shutting down http:          [OK]
```

- To restart the `httpd` web server daemon manually under Linux, you'll type:

```
[root@deep /]# /etc/init.d/httpd restart
Shutting down http:          [OK]
Starting httpd:              [OK]
```

Check inside your `/etc/init.d` directory for services available and use the commands **`start`** | **`stop`** | **`restart`** to work around. You will see along this book that we often use initialization script file to administration and control the way we start, restart, stop, etc services under Linux.

Software that must be uninstalled after installation of the server

Red Hat Linux installs other programs on your system by default and doesn't give you the choice to uninstall them during the install setup. For this reason, you must uninstall the following software after the complete installation of our Linux server.

Below is the list of programs and a short description of their purpose. We must uninstall them for increased security and to make more space on our server. For more information and an explanation of their capabilities and uses, please see your Red Hat manual or query the package by making an "**`rpm -qi foo`**" command before uninstalling them.

The `anacron` package:

The `anacron` package is similar to the `cron` package but differ in the way that it does not assume that the system is running continuously and it is a good command scheduler for system which doesn't run 24 hours a day. In server environment, the system should absolutely run 24/24 hours; therefore we simply don't need to have this kind of package installed on a server.

- To remove the **`anacron`** package from your system, use the following commands:

```
[root@deep /]# /etc/init.d/anacron stop
[root@deep /]# rpm -e anacron
[root@deep /]# rm -rf /var/spool/anacron/
```

The `apmd` package:

The `apmd` package or **A**dvanced **P**ower **M**anagement **D**aemon utilities is used on notebook computer. It can watch your notebook's battery and warn all users when the battery is low. As you can imagine, there is no need to have it installed on a server machine.

- To remove the **`apmd`** package from your system, use the following commands:

```
[root@deep /]# /etc/init.d/apmd stop
[root@deep /]# rpm -e apmd
```

The `at` package:

The `at` package is a utility that will do time-oriented job control by scheduling a command to run later. Unfortunately, it has had a rich history of problems and we can achieve the same functionality with the more secure `vixie-cron` package. For this reason I recommend you to uninstall it.

- To remove the **`at`** package from your system, use the following commands:

```
[root@deep /]# /etc/init.d/atd stop
[root@deep /]# rpm -e at
```

The `gpm` package:
The `gpm` package provides mouse support to text-based Linux applications. It's the software that allows you to cut-and-paste operations using the mouse on your terminal. If most of your entire administration of the server is made via remote connection, you can remove this package to save some processes and memories. We can continue to use cut-and-paste operation via remote connection to the server without problem. The `gpm` package is only useful if you stay at the console terminal of your server to make administration tasks.

- To remove the **`gpm`** package from your system, use the following commands:
```
[root@deep /]# /etc/init.d/gpm stop
[root@deep /]# rpm -e gpm
```

The `dhcpcd` package:
The `dhcpcd` package contains the protocol, which allows systems to get their own network configuration information from a `DHCP` server. If you are going to use `DHCP` on your network, it is recommended to install the `DHCP` client included in the `pump` package, which provides a faster and simpler `DHCP` client. For more information about `DHCP`, see its related chapter in this book.

- To remove the **`dhcpcd`** package from your system, use the following command:
```
[root@deep /]# rpm -e dhcpcd
```

The `eject` package:
The `eject` package contains an eject program that allows the user to eject removable media (typically CD-ROMs, floppy disks, Iomega Jaz or Zip disks) using software. This is an unneeded program to have installed in a server environment. You should keep it installed only if you're intended to run a tape backup on your system.

- To remove the **`eject`** package from your system, use the following command:
```
[root@deep /]# rpm -e eject
```

The `hotplug` package:
The `hotplug` package is a helper application for loading modules for `USB` devices. On a server environment, `USB` devices are not used at all and are required only on Linux workstation.

- To remove the **`hotplug`** package from your system, use the following command:
```
[root@deep /]# rpm -e hotplug
```

The `lokkit` package:
The `lokkit` package is a Firewall configuration application for an average end user and it is not designed to configure arbitrary firewalls since it is solely designed to handle typical dialup user and cable modem set-ups. It is not the answer to a complex firewall configuration, and it is not the equal of an expert firewall designer. Therefore remove it from your server and read the chapter related to `GIPTables` in this book for more information about secure firewall configuration.

- To remove the **`lokkit`** package from your system, use the following command:
```
[root@deep /]# rpm -e lokkit
```

The `ipchains` package:
The `ipchains` package is the old tool used with Linux kernel 2.2 for managing Linux kernel packet filtering capabilities and to set up firewalling on the network. A new and more powerful tool called "`IPTables`" exists and this is the one that we will use later in the book to set up our firewall on Linux.

- To remove the **`ipchains`** package from your system, use the following command:
 `[root@deep /]#` **`rpm -e ipchains`**

The `ksymoops` package:
The `ksymoops` package is a small tool used to report kernel oops and error message decoder. This package is useful for developers that work on the Linux kernel and want to debug or for users that want to report bugs with the kernel. The same result can be achieved with the `dmesg` command of Linux. Therefore, you can remove this package from your secure server.

- To remove the **`ksymoops`** package from your system, use the following command:
 `[root@deep /]#` **`rpm -e ksymoops`**

The `kudzu` package:
The `kudzu` package is a hardware-probing tool that runs at system boot time to determine what hardware has been added or removed from the system. Again, in server environment, we don't upgrade, add or remove hardware every time. Therefore, we can safety remove this small package from our server.

- To remove the **`kudzu`** package from your system, use the following command:
 `[root@deep /]#` **`rpm -e kudzu`**

The `mailcap` package:
`Metamail` is a program that uses the `mailcap` file to determine how it should display non-text or multimedia material. We don't need to have it installed at all. Remove it.

- To remove the **`mailcap`** package from your system, use the following command:
 `[root@deep /]#` **`rpm -e mailcap`**

The `pciutils` package:
The `pciutils` package contains various utilities for inspecting and setting devices connected to the `PCI` bus. Keep it installed if you want, but I recommend removing it form the server.

- To remove the **`pciutils`** package from your system, use the following command:
 `[root@deep /]#` **`rpm -e pciutils`**

The `raidtools` package:
The `raidtools` package includes the tools you need to set up and maintain a software `RAID` device on a Linux system. You should keep this package only if you have configured your server to run with `RAID` support. Otherwise, remove it.

- To remove the **`raidtools`** package from your system, use the following command:
 `[root@deep /]#` **`rpm -e raidtools`**

The `redhat-logos` package:
The `redhat-logos` package contains files of the Red Hat "Shadow Man" logo and the `RPM` logo.

- To remove the **`redhat-logos`** package from your system, use the following command:
 `[root@deep /]#` **`rpm -e redhat-logos`**

The `redhat-release` package:
The `redhat-release` package contains the Red Hat Linux release files. Please note that if you remove this package, the boot process of the system will generate error messages because it will look for a file called "`redhat-release`" which will not be available anymore. To solve this problem, we recreate the required file under the appropriated directory and add as content of this file the word "Red Hat Linux". Of course you can change it for whatever you want.

- To remove the **`redhat-release`** package from your system, use the command:
 `[root@deep /]#` **`rpm -e redhat-release`**
 `[root@deep /]#` **`echo Red Hat Linux > /etc/redhat-release`**

The `setserial` package:
The `setserial` package is a basic system utility for displaying or setting serial port information.

- To remove the **`setserial`** package from your system, use the command:
 `[root@deep /]#` **`rpm -e setserial`**

The `hdparm` package:
The program `hdparm` is needed by `IDE` hard disks but not `SCSI` hard disks. If you have an `IDE` disk on your system you must keep this program (`hdparm`), but if you don't have an `IDE` hard disk you can remove it safely from your system. `hdparm` is a small Linux tool used to optimize your `IDE` hard drive. `SCSI` hard drives don't need to be optimized since they are capable to run at their full speed (80 Mps to 160 Mps) without modification.

- To remove the **`hdparm`** package from your system, use the following command:
 `[root@deep /]#` **`rpm -e hdparm`**

The `mkinitrd` package:
The program `mkinitrd` is needed by `SCSI` or `RAID` hard disk but not `IDE` hard disks. If you have a `SCSI` or `RAID` disk on your system you must keep this program (`mkinitrd`), but if you don't have a `SCSI` or `RAID` hard disk you can safely remove it from your system.

- To remove the **`mkinitrd`** package from your system, use the following command:
 `[root@deep /]#` **`rpm -e --nodeps mkinitrd`**

The `xconfig` packages:
Use the programs `kbdconfig`, `mouseconfig`, `timeconfig`, `netconfig`, `authconfig`, `ntsysv`, and `setuptool` in order to set your keyboard language and type, your mouse type, your default time zone, your Ethernet devices, your `NIS` and shadow passwords, your numerous symbolic links in `/etc/rc.d` directory, and text mode menu utility which allow you to access all of these features.

After those configurations have been set during the installation stage of your Linux server it's rare that you would need to change them again. So, you can uninstall them, and if in the future you need to change your keyboard, mouse, default time, etc again via test mode menu, all you have to do is to install the program with the RPM from your original CD-ROM.

- To remove all the above programs from your system, use the following command:
  ```
  [root@deep /]# rpm -e kbdconfig mouseconfig timeconfig netconfig
  authconfig ntsysv setuptool
  ```

The newt package:
The newt package provides a development library for text mode user interfaces. It's mainly used by all the above config packages. Since all the config packages are removed from the server, we can safety remove newt from our system. If you have decided to keep the above config packages (kbdconfig, mouseconfig, etc), then you should keep newt, otherwise you should remove it.

- To remove the **newt** package from your system, use the following command:
  ```
  [root@deep /]# rpm -e newt
  ```

The lilo package:
The lilo package provides a boot loader for Linux. Remember that during our Linux installation, we have chosen to go with GRUB instead of LILO as our boot loader for Linux. Therefore, you can safety remove this package from your system since it's really not used now.

- To remove the **lilo** package from your system, use the following command:
  ```
  [root@deep /]# rpm -e lilo
  [root@deep /]# rm -f /etc/lilo.conf.anaconda
  ```

The reiserfs-utils package:
The reiserfs-utils package contains a number of utilities for creating, checking, modifying, and correcting any inconsistencies in ReiserFS file-systems. ReiserFS is another file-system like Ext3 for Linux. In our configuration of Linux we use Ext3 as our file-system therefore we don't need to keep this package installed. Keep this utility package only if you intended to run ReiserFS instead of Ext3 file-system on your Linux server.

- To remove the **reiserfs-utils** package from your system, use the command:
  ```
  [root@deep /]# rpm -e reiserfs-utils
  ```

The quota package:
The program quota is a system administration tools for monitoring and limiting user/group disk usage, per file system. This program must be installed only on servers where the need for monitoring and restricting amount of disk space in user's directories is require.

- To remove the **quota** package from your system, use the following command:
  ```
  [root@deep /]# rpm -e quota
  ```

The `indexhtml` package:
The `indexhtml` package contains the `HTML` page and graphics for a welcome page shown by your browser under graphical interface installation. These `HTML` pages are information about Red Hat software. You really don't need this package under server installation and especially when `GUI` is not available. Therefore, you can safety remove this package from your system.

- To remove the **`indexhtml`** package from your system, use the following command:
 `[root@deep /]#` **`rpm -e indexhtml`**

The `usbutils` package:
The `usbutils` package provides Linux `USB` utilities for inspecting devices connected to a `USB` bus on your system. In server environment, we really don't use any `USB` devices and can safety remove this package from our server installation. `USB` will usually be used only in Linux workstation installation where you want to plug printer, camera and any other media of this type.

- To remove the **`usbutils`** package from your system, use the following command:
 `[root@deep /]#` **`rpm -e usbutils`**

The `hwdata` package:
The `hwdata` package contains various hardware identification and configuration data mainly used with `USB` and `XFree86`. Remember that `XFree86` is related to graphical interface and since we don't use any `GUI` into our Linux server, we can remove this package from our system.

- To remove the **`hwdata`** package from your system, use the following command:
 `[root@deep /]#` **`rpm -e hwdata`**

The `parted` package:
The `parted` package contains various utilities to create, destroy, resize, move and copy hard disk partitions. It is useful when you need to play with your hard disk structure. In most case, we partition and set hard disk organization at the installation of the operating system and don't need to play or change something once everything are installed and running properly. It's rare to have to use this package utility on a production server and this is the reason why I recommend you to uninstall it. If you prefer to keep it installed for future possible usage, you are free to do it.

- To remove the **`parted`** package from your system, use the following command:
 `[root@deep /]#` **`rpm -e parted`**

The `hesiod` package:
The `hesiod` package is another one that we can uninstall from our Linux server configuration setup. It's a system which uses existing `DNS` functionality to provide access to databases of information that change infrequently. In most cases, we don't need it and you should keep it install only if you think that you will need it for some special configuration of your server.

- To remove the **`hesiod`** package from your system, use the following command:
 `[root@deep /]#` **`rpm -e hesiod`**

The `mt-st` package:
The `mt-st` package provides tools for controlling and managing tape drives on your system. You should keep it installed only if you have and want to run a tape backup media on your system.

- To remove the **`mt-st`** package from your system, use the following command:
```
[root@deep /]# rpm -e mt-st
```

The `man-pages` package:
The `man-pages` package provides a large collection of additional manual pages (man pages) from the **L**inux **D**ocumentation **P**roject (`LDP`). By default many manual pages are installed with the operating system and the `man-pages` package provides additional documentation for those who want to read them on the system. In server environment, I really don't see the need to have additional manual pages installed since these manual pages can be read online from the Internet or even on another computer running as a development or workstation.

- To remove the **`man-pages`** package from your system, use the following command:
```
[root@deep /]# rpm -e man-pages
```

The `sendmail` package:
Even if you don't want to run your system as a full mail server, mailer software is always needed for potential messages sent to the root user by different software services installed on your machine.

`Sendmail` is a **M**ail **T**ransport **A**gent (`MTA`) program that sends mail from one machine to another and it's the default mail server program installed on Red Hat Linux. Unfortunately, this software has a long history of security problem and for this reason I highly recommend you to not use it on your Linux server. You must uninstall `Sendmail` and see the part in this book that is related to **M**ail **T**ransfer **A**gent configuration and installation for some good alternative like `Exim` or `Qmail`.

- To remove the **`sendmail`** package from your system, use the following commands:
```
[root@deep /]# /etc/init.d/sendmail stop
[root@deep /]# rpm -e sendmail
```

The `procmail` package:
`Procmail` is a mail-processing program, which can be used by `Sendmail` for all local mail delivery and filtering. This program is required only if you decide to install and use `Sendmail` on your server and only if `Sendmail` is installed. Since we've decided to not go with `Sendmail` as our `MTA` for security reason, we can uninstall `procmail` from our Linux server.

- To remove the **`procmail`** package from your system, use the following command:
```
[root@deep /]# rpm -e procmail
```

The `openldap` package:
The `OpenLDAP` software is a set of protocols for accessing directory services like phone book style information and other kinds of information over the Internet. This useful program is not suitable for everyone and depends of what you want to do with your system. If you want to give it a try, see later in this book under the chapter related to databases for more information.

- To remove the **`OpenLDAP`** package from your system, use the following command:
```
[root@deep /]# rpm -e --nodeps openldap
```

The `cyrus-sasl` packages:
The `Cyrus SASL` implementation is the **S**imple **A**uthentication and **S**ecurity **L**ayer, a method for adding authentication support to connection-based protocols. It is used in conjunction with `Cyrus`, which is an electronic messaging program like `Sendmail`. Since `Cyrus SASL` is made to be used with `Sendmail` that we have removed previously for security reason, we can safety remove it.

- To remove the **`Cyrus SASL`** package from your system, use the following command:
 `[root@deep /]#` **`rpm -e --nodeps cyrus-sasl cyrus-sasl-md5 cyrus-sasl-plain`**

The `openssl` package:
`OpenSSL` is an `SSL` encryption mechanism which ensures and provides safe and secure transactions of data over networks. This piece of software is one of the most important tools for a Linux server and it is highly recommended that it is installed. Unfortunately, the one that comes with Red Hat Linux is not up to date and not optimized for our specific server. For this reason, we will uninstall it now and see later in this book, under the chapters related to security software, how to install, secure, optimize and use it.

- To remove the **`OpenSSL`** package from your system, use the following command:
 `[root@deep /]#` **`rpm -e --nodeps openssl`**
 `[root@deep /]#` **`rm -rf /usr/share/ssl/`**

The `ash` package:
The `ash` package is a smaller version of the bourne **sh**ell (`sh`). Since we already use `sh`, we can uninstall this package from our system. If you use this program in your regular administration task, then keep it installed on your server. In most case, we can remove it.

- To remove the **`ash`** package from your system, use the following command:
 `[root@deep /]#` **`rpm -e ash`**

The `tcsh` package:
The `tcsh` package is an enhanced version of `csh`, another `C` shell. We already have `bash` as our default shell program on Linux, and I don't find any reason to keep another variant installed if we don't have any program or services that need it to run.

Most services under Linux can easily run with our default `bash` shell program and if you don't have any program that required `tcsh` to run, then I recommend you to uninstall it. If in the future, you see that you need to have `tcsh` installed on your server for some specific program to run, then all you have to do is to install it from your CD-ROM. In most cases, there is no program that needs `tsch` to run, therefore you can remove it.

- To remove the **`tcsh`** package from your system, use the following command:
 `[root@deep /]#` **`rpm -e tcsh`**

The `specspo` package:
The `specspo` package contains the portable object catalogues used to internationalize Red Hat packages. I don't think that this kind of package is really required on a production server.

- To remove the **`specspo`** package from your system, use the following command:
```
[root@deep /]# rpm -e specspo
```

The `krb5-libs` package:
The `krb5-libs` package contains the shared libraries needed by `Kerberos 5`. Because we're not using `Kerberos`, we'll need to uninstall this package. `Kerberos` is not secure as you can think and can be cracked easily with some good knowledge of this program. Anyway it is yours to decide if you really need it.

- To remove the **`krb5-libs`** package from your system, use the following command:
```
[root@deep /]# rpm -e krb5-libs
[root@deep /]# rm -rf /usr/kerberos/
```

The `MAKEDEV` package:
The `MAKEDEV` package contains the `MAKEDEV` program, which makes it easier to create and maintain the files in the `/dev` directory. Program provided by this package is used for creating device files in the `/dev` directory of your server. In general, we use it under development server when we build new package for our Linux system. On production servers, it's rare to use it. Therefore we can remove it from our system without any problem.

- To remove the **`MAKEDEV`** package from your system, use the following command:
```
[root@deep /]# rpm -e MAKEDEV
```

Remove unnecessary documentation files
By default the majority of each `RPM`'s packages installed under Linux come with documentation files related to the software. This documentation contains original files from the program tar archive like `README`, `FAQ`, `BUG`, `INSTALL`, `NEWS`, `PROJECTS` and more.

Many of them can be easily retrieved from the website where the program has been downloaded and it makes no sense for them to be kept on your system. I know that hard drives costs have come down considerably recently, but why keep this kind of documentation on a secure server if it unlikely they will not be read more than once. Anyway, have a look inside those files and decide for yourself if you want to remove them or not.

- To remove all documentation files from your system, use the following commands:
```
[root@deep /]# cd /usr/share/doc/
[root@deep doc]# rm -rf *
```

Remove unnecessary/empty files and directories

There are some files and directories we can remove manually from the file system of Linux to make a clean install. These files and directories are not needed but still exist after our secure installation of Linux and can be removed safely. Some are bugs from the Red Hat installation script and others are created by default even if you don't use them.

- To remove all unnecessary files and directories from your system, use the commands:

```
[root@deep /]# rm -f  /etc/exports
[root@deep /]# rm -f  /etc/printcap
[root@deep /]# rm -f  /etc/ldap.conf
[root@deep /]# rm -f  /etc/krb.conf
[root@deep /]# rm -f  /etc/yp.conf
[root@deep /]# rm -f  /etc/hosts.allow
[root@deep /]# rm -f  /etc/hosts.deny
[root@deep /]# rm -f  /etc/csh.login
[root@deep /]# rm -f  /etc/csh.cshrc
[root@deep /]# rm -f  /etc/fstab.REVOKE
[root@deep /]# rm -f  /etc/pam_smb.conf
[root@deep /]# rm -rf /etc/xinetd.d/
[root@deep /]# rm -rf /etc/opt/
[root@deep /]# rm -rf /etc/X11/
[root@deep /]# rm -rf opt/
[root@deep /]# rm -rf /var/opt/
[root@deep /]# rm -rf /var/nis/
[root@deep /]# rm -rf /var/yp/
[root@deep /]# rm -rf /var/lib/games/
[root@deep /]# rm -rf /var/spool/lpd/
[root@deep /]# rm -rf /usr/lib/python1.5/
[root@deep /]# rm -rf /usr/lib/games/
[root@deep /]# rm -rf /usr/X11R6/
[root@deep /]# rm -rf /usr/etc/
[root@deep /]# rm -rf /usr/games/
[root@deep /]# rm -rf /usr/local/
[root@deep /]# rm -rf /usr/dict/
[root@deep /]# rm -f  /usr/bin/X11
[root@deep /]# rm -f  /usr/lib/X11
```

NOTE: If in the future you want to install a program which needs some of the files/directories we have removed, then the program will automatically recreate the missing files or directories. Good!

Software that must be installed after installation of the server

There are certain programs required to be able to compile programs on your server, for this reason you must install the following `RPM` packages. This part of the installation is very important and requires that you install all the packages described below.

These are on your Red Hat Part 1 and Part 2 CD-ROMs under `RedHat/RPMS` directory and represent the necessary base software needed by Linux to compile and install programs. Please note that if you don't want to compile software in your server or if you only use `RPM`'s packages to update programs or if you use a dedicated server to develop, compile or create your own `RPM`'s packages which will be installed later along your network on the servers, then you **DON'T** need to install the packages described here.

Step 1
First, we mount the CD-ROM drive and move to the RPMS subdirectory of the CD-ROM.

- To mount the CD-ROM drive and move to RPM directory, use the following commands:

```
[root@deep /]# mount /dev/cdrom /mnt/cdrom/
had: ATAPI 32X CD-ROM drive, 128kB Cache
mount: block device dev/cdrom is write-protected, mounting read-only
[root@deep /]# cd /mnt/cdrom/RedHat/RPMS/
```

These are the packages that we need to be able to compile and install programs on the Linux system. Remember, this is the minimum number of packages that permits you to compile most of the tarballs available for Linux. Other compiler packages exist on the Linux CD-ROM, so verify with the README file that came with the tarballs program you want to install if you receive error messages during compilation of the specific software.

The compiler packages:
Compiler packages contain programs and languages used to build software on the system. Remember to uninstall the entire following compiler packages after successful installation of all software required on your Linux server.

binutils-2.11.93.0.2-11.i386.rpm	flex-2.5.4a-23.i386.rpm
bison-1.35-1.i386.rpm	gcc-2.96-110.i386.rpm
byacc-1.9-19.i386.rpm	gcc-c++-2.96-110.i386.rpm
cdecl-2.5-22.i386.rpm	glibc-kernheaders-2.4-7.14.i386.rpm
cpp-2.96-110.i386.rpm	m4-1.4.1-7.i386.rpm
cproto-4.6-9.i386.rpm	make-3.79.1-8.i386.rpm
ctags-5.2.2-2.i386.rpm	patch-2.5.4-12.i386.rpm
dev86-0.15.5-1.i386.rpm	perl-5.6.1-34.99.6.i386.rpm

The development packages:
Development packages contain header and other files required during compilation of software. In general, development packages are needed when we want to add some specific functionality to the program that we want to compile. For example if I want to add PAM support to IMAP, I'll need pam-devel, which contains the required header files for IMAP to compile successfully.

As for compiler packages, all development packages must be uninstalled after successful compilation of all the software that you need on your Linux server. Remember to uninstall them since they are not needed for proper functionality of the server, but just to compile the programs.

aspell-devel-0.33.7.1-9.i386.rpm	libpng-devel-1.0.12-2.i386.rpm
db3-devel-3.3.11-6.i386.rpm	libstdc++-devel-2.96-110.i386.rpm
freetype-devel-2.0.9-2.i386.rpm	ncurses-devel-5.2-26.i386.rpm
gdbm-devel-1.8.0-14.i386.rpm	pam-devel-0.75-32.i386.rpm
gd-devel-1.8.4-4.i386.rpm	pspell-devel-0.12.2-8.i386.rpm
glibc-devel-2.2.5-34.i386.rpm	zlib-devel-1.1.3-25.7.i386.rpm
libjpeg-devel-6b-19.i386.rpm	

Dependencies packages:
Dependencies packages are other RPM packages needed by the RPM packages that we want to install. This happens because some RPM's are directly linked with others and depend on each one to function properly. The following packages are required by the above RPM packages and we will install them to satisfy dependencies. After proper compilation and installation of all needed software on the Linux server, we can uninstall them safety (if not needed by special program that we will install).

aspell-0.33.7.1-9.i386.rpm	libpng-1.0.12-2.i386.rpm
freetype-2.0.9-2.i386.rpm	libtool-libs-1.4.2-7.i386.rpm
gd-1.8.4-4.i386.rpm	pspell-0.12.2-8.i386.rpm
libjpeg-6b-19.i386.rpm	

Step 2
It is better to install the software described above together if you don't want to receive dependencies error messages during the install. Some of the RPMs reside on CD-ROM Part 1 and other on CD-ROM Part2 of Red Hat. For easy installation, I recommend you to copy all of the required packages (compilers and development) to your hard drive and install them from there.

- These procedures can be accomplished with the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rpm -Uvh *.rpm
Preparing...                ########################################### [100%]
   1:binutils               ########################################### [  2%]
   2:bison                  ########################################### [  5%]
   3:byacc                  ########################################### [  8%]
   4:cdecl                  ########################################### [ 11%]
   5:cpp                    ########################################### [ 13%]
   6:cproto                 ########################################### [ 16%]
   7:ctags                  ########################################### [ 19%]
   8:db3-devel              ########################################### [ 22%]
   9:dev86                  ########################################### [ 25%]
  10:flex                   ########################################### [ 27%]
  11:freetype               ########################################### [ 30%]
  12:freetype-devel         ########################################### [ 33%]
  13:gdbm-devel             ########################################### [ 36%]
  14:glibc-kernheaders      ########################################### [ 38%]
  15:glibc-devel            ########################################### [ 41%]
  16:gcc                    ########################################### [ 44%]
  17:libjpeg                ########################################### [ 47%]
  18:libjpeg-devel          ########################################### [ 50%]
  19:libpng                 ########################################### [ 52%]
  20:gd                     ########################################### [ 55%]
  21:gd-devel               ########################################### [ 58%]
  22:libstdc++-devel        ########################################### [ 61%]
  23:gcc-c++                ########################################### [ 63%]
  24:libtool-libs           ########################################### [ 66%]
  25:m4                     ########################################### [ 69%]
  26:make                   ########################################### [ 72%]
  27:pam-devel              ########################################### [ 75%]
  28:patch                  ########################################### [ 77%]
  29:perl                   ########################################### [ 80%]
  30:ncurses-devel          ########################################### [ 83%]
  31:pspell                 ########################################### [ 86%]
  32:aspell                 ########################################### [ 88%]
  33:pspell-devel           ########################################### [ 91%]
  34:aspell-devel           ########################################### [ 94%]
  35:zlib-devel             ########################################### [ 97%]
  36:libpng-devel           ########################################### [100%]
```

Step 3
After installation and compilation of all programs and services, it's a good idea to remove all sharp objects (compilers, etc) described above unless they are required by your system.

A few reasons are:

- ✓ If a cracker gains access to your server he or she cannot compile or modify binary programs. Also, this will free a lot of space and will help to improve regular scanning of the files on your server for integrity checking.

- ✓ When you run a server, you will give it a special task to accomplish. You will never put all services you want to offer in one machine or you will lose speed (resources available divided by the number of process running on the server).

- ✓ Decrease your security with a lot of services running on the same machine, if a cracker accesses this server, he or she can attack directly all the others available.

- ✓ Having different servers doing different tasks will simplify the administration, and management. You know what task each server is supposed to do, what services should be available, which ports are open to clients access and which one are closed, you know what you are supposed to see in the log files, etc, and give you more control and flexibility on each one (server dedicated for mail, web pages, database, development, backup, etc.

- ✓ For example, one server specialized just for development and testing will mean you will not be compelled to install compiler programs on a server each time you want to compile and install new software on it, and be obliged afterwards to uninstall the compilers, or other sharp objects.



Part II

System Security & Optimization

CHAPTER 3

General Security

IN THIS CHAPTER

1. **BIOS**
2. **Unplug your server from the network**
3. **Security as a policy**
4. **Choose a right password**
5. **The root account**
6. **Set login time out for the root account**
7. **Shell logging**
8. **The single-user login mode of Linux**
9. **Disabling Ctrl-Alt-Delete keyboard shutdown command**
10. **Limiting the default number of started ttys on the server**
11. **The LILO and /etc/lilo.conf file**
12. **The GRUB and /boot/grub/grub.conf file**
13. **The /etc/services file**
14. **The /etc/securetty file**
15. **Special accounts**
16. **Control mounting a file system**
17. **Mounting the /usr directory of Linux as read-only**
18. **Tighten scripts under /etc/init.d/**
19. **Tighten scripts under /etc/cron.daily**
20. **Bits from root-owned programs**
21. **Don’t let internal machines tell the server what their MAC address is**
22. **Unusual or hidden files**
23. **Finding Group and World Writable files and directories**
24. **Unowned files**
25. **Finding .rhosts files**
26. **Physical hard copies of all-important logs**
27. **Getting some more security by removing manual pages**
28. **System is compromised!**

Linux General Security

Abstract

A secure Linux server depends on how the administrator makes it. Once we have eliminated the potential security risk by removing unneeded services, we can start to secure our existing services and software on our server. Within a few hours of installing and configuring your system, you can prevent many attacks before they occur. In this chapter we will discuss some of the more general, basic techniques used to secure your system. The following is a list of features that can be used to help prevent attacks from external and internal sources.

BIOS

It is recommended to disallow booting from floppy drives and set passwords on BIOS features. You can check your BIOS manual or look at it thoroughly the next time you boot up your system to find out how to do this. Disabling the ability to boot from floppy drives and being able to set a password to access the BIOS features will improve the security of your system.

This will block unauthorized people from trying to boot your Linux system with a special boot disk and will protect you from people trying to change BIOS features like allowing boot from floppy drive or booting the server without prompt password. It is important to note that there is a possibility to bypass this security measure if someone has a physical access to your server since they can open the computer and unplug the BIOS battery. This will reset all features to their initial values.

Unplug your server from the network

It is not wise to apply security changes in your newly installed Linux server if you are online. So it is preferable to deactivate all network interfaces in the system before applying security changes.

- To stop specific network devices manually on your system, use the command:
```
[root@deep /]# ifdown eth0
```
- To start specific network devices manually on your system, use the command:
```
[root@deep /]# ifup eth0
```
- To shut down all network interfaces, use the following command:
```
[root@deep /]# /etc/init.d/network stop
Shutting down interface eth0          [OK]
Disabling Ipv4 packet forwarding      [OK]
```
- To start all network interfaces, use the following command:
```
[root@deep /]# /etc/init.d/network start
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
```

Security as a policy

It is important to point out that you cannot implement security if you have not decided what needs to be protected, and from whom. You need a security policy--a list of what you consider allowable and what you do not consider allowable upon which to base any decisions regarding security. The policy should also determine your response to security violations. What you should consider when compiling a security policy will depend entirely on your definition of security. The following questions should provide some general guidelines:

- ✓ How do you classify confidential or sensitive information?
- ✓ Does the system contain confidential or sensitive information?
- ✓ Exactly whom do you want to guard against?
- ✓ Do remote users really need access to your system?
- ✓ Do passwords or encryption provide enough protection?
- ✓ Do you need access to the Internet?
- ✓ How much access do you want to allow to your system from the Internet?
- ✓ What action will you take if you discover a breach in your security?

This list is short, and your policy will probably encompass a lot more before it is completed. Any security policy must be based on some degree of paranoia; deciding how much you trust people, both inside and outside your organization. The policy must, however, provide a balance between allowing your users reasonable access to the information they require to do their jobs and totally disallowing access to your information. The point where this line is drawn will determine your policy.

Choose a right password

The starting point of our Linux general security tour is the password. Many people keep their valuable information and files on a computer, and the only thing preventing others from seeing it is the eight-character string called a password. An unbreakable password, contrary to popular belief, does not exist. Given time and resources all passwords can be guessed either by social engineering or by brute force.

Social engineering of server passwords and other access methods are still the easiest and most popular way to gain access to accounts and servers. Often, something as simple as acting as a superior or executive in a company and yelling at the right person at the right time of the day yields terrific results.

Running a password cracker on a weekly basis on your system is a good idea. This helps to find and replace passwords that are easily guessed or weak. Also, a password checking mechanism should be present to reject a weak password when choosing an initial password or changing an old one. Character strings that are plain dictionary words, or are all in the same case, or do not contain numbers or special characters should not be accepted as a new password.

We recommend the following rules to make passwords effective:

- ✓ They should be at least six characters in length, preferably eight characters including at least one numeral or special character.

- ✓ They must not be trivial; a trivial password is one that is easy to guess and is usually based on the user's name, family, occupation or some other personal characteristic.

- ✓ They should have an aging period, requiring a new password to be chosen within a specific time frame.

- ✓ They should be revoked and reset after a limited number of concurrent incorrect retries.

The root account

The "root" account is the most privileged account on a Unix system. The "root" account has no security restrictions imposed upon it. This means the system assumes you know what you are doing, and will do exactly what you request -- no questions asked. Therefore it is easy, with a mistyped command, to wipe out crucial system files. When using this account it is important to be as careful as possible. For security reasons, never log in on your server as "root" unless it is absolutely an instance that necessitates root access. Also, if you are not on your server, never sign in and leave yourself on as "root"--this is VERY, VERY. VERY BAD.

Set login time out for the root account

Despite the notice to never, if they are not on the server, sign in as "root" and leave it unattended, administrators still stay on as "root" or forget to logout after finishing their work and leave their terminals unattended.

Step 1
The answer to solve this problem is to make the bash shell automatically logout after not being used for a period of time. To do that, you must set the special variable of Linux named "TMOUT" to the time in seconds of no input before logout.

- Edit your **profile** file (vi /etc/profile) and add the following line somewhere after the line that read "HISTSIZE=" on this file:

```
HOSTNAME=`/bin/hostname`
HISTSIZE=1000
TMOUT=7200
```

The value we enter for the variable "TMOUT=" is in seconds and represents 2 hours (60 * 60 = 3600 * 2 = 7200 seconds). It is important to note that if you decide to put the above line in your /etc/profile file, then the automatic logout after two hours of inactivity will apply for all users on the system. So, instead, if you prefer to control which users will be automatically logged out and which ones are not, you can set this variable in their individual .bashrc file.

Step 2
Once we have added the above line to the profile file, we must add its definition to the export line of the same file as follow.

- Edit your **profile** file (vi /etc/profile) and change the line:

```
export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE INPUTRC
```

To read:

```
export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE TMOUT INPUTRC
```

After this parameter has been set on your system, you must logout and login again (as root) for the change to take effect.

Shell logging

To make it easy for you to repeat long commands, the bash shell stores up to 500 old commands in the `~/.bash_history` file (where "**~/**" is your home directory). Each user that has an account on the system will have this file `.bash_history` in their home directory. Reducing the number of old commands the `.bash_history` files can hold may protect users on the server who enter by mistake their password on the screen in plain text and have their password stored for a long time in the `.bash_history` file.

Step 1
The `HISTSIZE` line in the `/etc/profile` file determine the size of old commands the `.bash_history` file for all users on your system can hold. For all accounts I would highly recommend setting the `HISTSIZE` in `/etc/profile` file to a low value such as **`10`**.

- Edit the **`profile`** file (`vi /etc/profile`) and change the line:

 `HISTSIZE=1000`

 To read:

 `HISTSIZE=10`

This means, the `.bash_history` file in each user's home directory can store 10 old commands and no more. Now, if a cracker tries to see the `~/.bash_history` file of users on your server to find some password typed by mistake in plain text, he or she has less chance to find one.

Step 2
The administrator should also add into the `/etc/profile` file the "`HISTFILESIZE=0`" line, so that each time a user logs out, its `.bash_history` file will be deleted so crackers will not be able to use `.bash_history` file of users who are not presently logged into the system.

- Edit the **`profile`** file (`vi /etc/profile`) and add the following parameter below the "`HISTSIZE=`" line:

 `HISTFILESIZE=0`

Step 3
Once we have added the above line to the `profile` file, we must add its definition to the `export` line of the same file as follow.

- Edit your **`profile`** file (`vi /etc/profile`) and change the line:

 `export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE TMOUT INPUTRC`

 To read:

 `export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE` **`HISTFILESIZE`** `TMOUT INPUTRC`

After this parameter has been set on your system, you must logout and login again (as root) for the change to take effect.

The single-user login mode of Linux

This part applies for those who use `LILO` as their boot loader. Linux has a special command (`linux single`) also known as 'single-user mode', which can be entered at the boot prompt during startup of the system. The single-user mode is generally used for system maintenance.

You can boot Linux in single-user mode by typing at the `LILO` boot prompt the command:

```
LILO:  linux single
```

This will place the system in Run level 1 where you'll be logged in as the super-user '`root`', and where you won't even have to type in a password!

Step 1
Requiring no password to boot into root under single-user mode is a bad idea! You can fix this by editing the **`inittab`** file (`vi /etc/inittab`) and change the following line:

```
id:3:initdefault:
```

To read:

```
id:3:initdefault:
~~:S:wait:/sbin/sulogin
```

The addition of the above line will require entering the root password before continuing to boot into `single-user` mode by making `init (8)` run the program `sulogin (8)` before dropping the machine into a root shell for maintenance.

Step 2
Now, we have to restart the process control initialization of the server for the changes to take effect.

- This can be done with the following command:
 `[root@deep /]# /sbin/init q`

Disabling `Ctrl-Alt-Delete` keyboard shutdown command

Commenting out the line (with a "**#**") listed below in your `/etc/inittab` file will disable the possibility of using the `Control-Alt-Delete` command to shutdown your computer. This is pretty important if you don't have the best physical security to the machine.

Step 1

- To do this, edit the **`inittab`** file (`vi /etc/inittab`) and change/comment the line:

  ```
  ca::ctrlaltdel:/sbin/shutdown -t3 -r now
  ```

 To read:

  ```
  #ca::ctrlaltdel:/sbin/shutdown -t3 -r now
  ```

Step 2
Now, we have to restart the process control initialization of the server for the changes to take effect.

- This can be done with the following command:
 `[root@deep /]# /sbin/init q`

Limiting the default number of started `ttys` on the server

Default installed Linux system comes with six virtual consoles in standard run levels. This means that six `mingetty` processes will always by available at every time on the server. These virtual consoles allow you to login with six different virtual consoles on the system.

Step 1
On secure server, we can limit the number to two virtual consoles and save some resources which may be used for other work by the server when required.

- To do this, edit the **`inittab`** file (`vi /etc/inittab`) and remove/comment the lines:

```
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
```

To read:

```
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
#3:2345:respawn:/sbin/mingetty tty3
#4:2345:respawn:/sbin/mingetty tty4
#5:2345:respawn:/sbin/mingetty tty5
#6:2345:respawn:/sbin/mingetty tty6
```

Step 2
Now, we have to restart the process control initialization of the server for the changes to take effect.

- This can be done with the following command:
 `[root@deep /]# /sbin/init q`

The `LILO` and `/etc/lilo.conf` file

This part applies for those who use `LILO` as their boot loader. `LILO` is a boot loader that can be used to manage the boot process, boot Linux kernel images from floppy disks, hard disks or can even act as a "boot manager" for other operating systems.

`LILO` is very important in the Linux system and for this reason, we must protect it the best we can. The most important configuration file of `LILO` is the `lilo.conf` file. It is with this file that we can configure and improve the security of our `LILO` program and Linux system. Following are three important options that will improve the security of our valuable `LILO` program.

- Adding: **timeout=00**

This option controls how long (in seconds) LILO waits for user input before booting to the default selection. One of the requirements of C2 security is that this interval be set to 0 unless the system dual boots something else.

- Adding: **restricted**

This option asks for a password only, if parameters are specified on the command line (e.g. linux single). The option "restricted" can only be used together with the "password" option. Make sure you use this one on each additional image you may have.

- Adding: **password=<password>**

This option asks the user for a password when trying to load the image. Actually the effect of using the password parameter in /etc/lilo.conf will protect the Linux image from booting. This means, it doesn't matter if you load Linux in single mode or if you just do a normal boot. It will always ask you for the password.

Now this can have a very bad effect, namely you are not able to reboot Linux remotely any more since it won't come up until you type in the root password at the console. It is for this reason that adding "restricted" with "password" is very important since the option "restricted" relaxes the password protection and a password is required only if parameters are specified at the LILO prompt, (e.g. single).

Passwords are always case-sensitive, also make sure the /etc/lilo.conf file is no longer world readable, or any user will be able to read the password. Here is an example of our protected LILO with the lilo.conf file.

Step 1

- Edit the **lilo.conf** file (vi /etc/lilo.conf) and add or change the three options above as show:

```
boot=/dev/sda
map=/boot/map
install=/boot/boot.b
prompt ← remove this line if you don't want to pass options at the LILO prompt.
timeout=00 ← change this line to 00 to disable the LILO prompt.
linear
message=/boot/message ← remove this line if you don't want the welcome screen.
default=linux
restricted ← add this line to relaxes the password protection.
password=<password> ← add this line and put your password.

image=/boot/vmlinuz-2.4.2-2
    label=linux
    initrd=/boot/initrd-2.4.2-2.img
    read-only
    root=/dev/sda6
```

Step 2

Because the configuration file /etc/lilo.conf now contains unencrypted passwords, it should only be readable for the super-user "root".

- To make the /etc/lilo.conf file readable only by the super-user "root", use the following command:

```
[root@deep /]# chmod 600 /etc/lilo.conf (will be no longer world readable).
```

Step 3
Now we must update our configuration file `/etc/lilo.conf` for the change to take effect.

- To update the `/etc/lilo.conf` file, use the following command:

```
[root@deep /]# /sbin/lilo -v
LILO version 21.4-4, copyright © 1992-1998 Wernerr Almesberger
'lba32' extentions copyright © 1999,2000 John Coffman

Reading boot sector from /dev/sda
had : ATAPI 32X CD-ROM drive, 128kB Cache
Merging with /boot/boot.b
Mapping message file /boot/message
Boot image : /boot/vmlinuz-2.2.16-22
Mapping RAM disk /boot/initrd-2.2.16-22.img
Added linux *
/boot/boot.0800 exists - no backup copy made.
Writing boot sector.
```

Step 4
One more security measure you can take to secure the `lilo.conf` file is to set it immutable, using the **`chattr`** command.

- To set the file immutable simply, use the following command:

```
[root@deep /]# chattr +i /etc/lilo.conf
```

And this will prevent any changes (accidental or otherwise) to the `lilo.conf` file. If you wish to modify the `lilo.conf` file you will need to unset the immutable flag:

- To unset the immutable flag, use the following command:

```
[root@deep /]# chattr -i /etc/lilo.conf
```

WARNING: When you use the `password` option, then `LILO` will always ask you for the password, regardless if you pass options at the `LILO` prompt (e.g. `single`) or not **EXCEPT** when you set the "`restricted`" option in `/etc/lilo.conf`.

The option "`restricted`" relaxes the password protection and a password is required only if parameters are specified at the `LILO` prompt, (e.g. `single`).

If you didn't had this option set "`restricted`", Linux will always ask you for the password and you will not be able to remotely reboot your system, therefore don't forget to add the option "`restricted`" with the option "`password`" into the `/etc/lilo.conf` file.

The `GRUB` and `/boot/grub/grub.conf` file

This part applies for those who use `GRUB` as their boot loader. `GRUB` is another boot loader like `LILO` but with many more useful feature and power. One of its main advantages compared to `LILO` is the fact that it provides a small shell interface to manage the operating system. Also, it doesn't need to be updated each time you recompile a new kernel on your server.

GRUB is very important since it is the first software program that runs when the computer starts and we have to secure it as much as possible to avoid any possible problem. In its default installation it's already well protected and below I explain how its configuration file is made. In regard to LILO, GRUB is really easy to use and configure. Below is a default GRUB configuration file and security I recommend you to apply. The text in bold are the parts of the configuration file that must be customized and adjusted to satisfy our needs.

- Edit the **grub.conf** file (vi /boot/grub/grub.conf) and set your needs. Below is what we recommend you:

```
default=0
timeout=0
splashimage=(hd0,0)/grub/splash.xpm.gz
password --md5 $1$oKr0ÝmFo$tPYwkkvQbtqo1erwHj5wb/
title Red Hat Linux (2.4.18-3)
        root (hd0,0)
        kernel /vmlinuz-2.4.18-3 ro root=/dev/sda5
        initrd /initrd-2.4.18-3.img
```

This tells the grub.conf file to set itself up for this particular configuration with:

default=0
The option "default" is used to define the default entry of the configuration file. The number "0" mean that the following parameters are the default entry for the configuration of GRUB. In a server configuration where Linux is the only operating system to boot, the default entry of "0" will be the only one to use and we don't need to define any additional entry.

timeout=0
This option "timeout" is used to define the timeout, in sec seconds to wait, before automatically booting the default entry. As for LILO, one of the requirements of C2 security is that this interval be set to 0 unless the system dual boots something else. One of the disadvantages to set this option to "0" is that you will no longer be able to have access at boot time to the shell interface of the software but this is not really a problem since all we need from the GRUB software is to boot our operating system.

splashimage=(hd0,0)/grub/splash.xpm.gz
This option "splashimage" is an option added by Red Hat to boot the system with a graphical image. The value is the path of the compressed image to use when booting GRUB. It's to you to keep this parameter on your system or to remove it. If you want to remove it, just delete the above line with the compressed image from your server.

password --md5 1bgGCL/$4yF3t0py.IjU0LU.q7YfB1
This option "password" is used to inform GRUB to ask for a password and disallows any interactive control, until you press the key <p> and enter a correct password. The option --md5 tells GRUB that a password in MD5 format is required as a value. If it is omitted, GRUB assumes the specified password is in clear text.

When we have installed the operating system, we have already configured GRUB with a password protection. This password is what you see here. If you want to change it, you have to use the "grub-md5-crypt" command to generate a new encrypt password it in MD5 format.

- This can be done with the following command:

```
[root@dev /]# grub-md5-crypt
Password:
$1$bgGCL/$4yF3t0py.IjU0LU.q7YfB1
```

Once the above command has been issued, you have to cut and paste the encrypted password to your configuration file.

`title Red Hat Linux (2.4.18-3)`
This option "`title`" is used to define a name to the contents of the rest of the line. It is directly related to the default boot entry. What you enter here is what you will see during boot time. This option is useful when we have more than one OS to start on our computer and allow us to give the name that we want to distinguish them. You are free to enter whatever name you like.

`root (hd0,0)`
This option "`root`" is one of the most important parameter with `GRUB` and without it nothing will work. It is used to define the current root device to use for booting the operating system. Its definition and configuration is a little bit special as you can see. Here is an explanation of its meaning. The "`hd0`" parameter represents using the entire disk and the "`hd0,0`" represents using the partition of the disk (or the boot sector of the partition when installing `GRUB`). Don't be confused here because "`hd`" is valid for `IDE` and `SCSI` drives. There is no difference; you always use "`hd`" even on `SCSI` drive.

`kernel /vmlinuz-2.4.18-3 ro root=/dev/sda5`
This option "`kernel`" is used to load the primary boot image (our kernel). The parameter to this option is simply the path where `GRUB` should find the kernel image from which we want it to boot. The additional lines are to inform it that kernel image is located on the `sda5` partition on our server and that we want to load it as read only for security reason.

`initrd /initrd-2.4.18-3.img`
This option "`initrd`" is optional and will appear into your `GRUB` configuration file only if you run a `SCSI` computer. For `IDE` computer, this option is not required and should not be defined inside the configuration file of `GRUB`. The parameter simply informs `GRUB` software where our initial ram disk image is located on the server. `GRUB` reads this initial ram disk and loads it during startup.

The `/etc/services` file

The port numbers on which certain "standard" services are offered are defined in the RFC 1700 "Assigned Numbers". The `/etc/services` file enables server and client programs to convert service names to these numbers (ports). The list is kept on each host and it is stored in the file `/etc/services`. Only the "`root`" user is allowed to make modifications to this file. It is rare to edit the `/etc/services` file since it already contains the more common service names / port numbers. To improve security, we can set the immutable flag on this file to prevent unauthorized deletion or modification.

- To immunize the `/etc/services` file, use the following command:
 `[root@deep /]#` **`chattr +i /etc/services`**

The `/etc/securetty` file

The `/etc/securetty` file allows you to specify which **`TTY`** and **`VC`** (virtual console) devices the "`root`" user is allowed to login on. The `/etc/securetty` file is read by the login program (usually `/bin/login`). Its format is a list of the **`tty`** and **`vc`** devices names allowed, and for all others that are commented out or do not appear in this file, `root` login is disallowed.

Disable any **`tty`** and **`vc`** devices that you do not need by commenting them out (**`#`** at the beginning of the line) or by removing them.

- Edit the **`securetty`** file (`vi /etc/securetty`) and comment out or remove the lines:

```
vc/1                                    tty1
#vc/2                                   #tty2
#vc/3                                   #tty3
#vc/4                                   #tty4
#vc/5                                   #tty5
#vc/6                                   #tty6
#vc/7                                   #tty7
#vc/8                                   #tty8
#vc/9                                   #tty9
#vc/10                                  #tty10
#vc/11                                  #tty11
```

Which means `root` is allowed to login on only **`tty1`** and **`vc/1`**. This is my recommendation, allowing "`root`" to log in on only one `tty` or `vc` device and use the `su` or `sudo` command to switch to "`root`" if you need more devices to log in as "`root`".

Special accounts

It is important to **DISABLE ALL default vendor accounts that** you don't use on your system (some accounts exist by default even if you have not installed the related services on your server). This should be checked after each upgrade or new software installation. Linux provides these accounts for various system activities, which you may not need if the services are not installed on your server. If you do not need the accounts, remove them. The more accounts you have, the easier it is to access your system.

We assume that you are using the shadow password suite on your Linux system. If you are not, you should consider doing so, as it helps to tighten up security somewhat. This is already set if you've followed our Linux installation procedure and selected, under the "Authentication Configuration", the option to "Enable Shadow Passwords" (see the chapter related to the "Installation of your Linux Server" for more information).

- To delete user on your system, use the following command:
 `[root@deep /]# userdel username`

- To delete group on your system, use the following command:
 `[root@deep /]# groupdel username`

Step 1
First we will remove all default vendor accounts into the `/etc/passwd` file that are unnecessary for the operation of the secure server configuration that we use in this book.

- Type the following commands to delete all default users accounts listed below.

```
[root@deep /]# userdel adm
[root@deep /]# userdel lp
[root@deep /]# userdel shutdown
[root@deep /]# userdel halt
[root@deep /]# userdel news
[root@deep /]# userdel mailnull
[root@deep /]# userdel operator
[root@deep /]# userdel games
[root@deep /]# userdel gopher
[root@deep /]# userdel ftp
[root@deep /]# userdel vcsa
```

WARNING: By default, the `userdel` command will not delete a user's home directory. If you want the home directories of accounts to be deleted too, then add the **`-r`** option to the `userdel` command. Finally, the **`-r`** option must be used only when you have added a new user to the server and want to remove them. It doesn't need to be used for the removal of the above default user's accounts since they do not have a home directory.

Once the above list of users has been deleted from your Linux system, the `/etc/passwd` file will look like this:

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
uucp:x:10:14:uucp:/var/spool/mail:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
rpm:x:37:37::/var/lib/rpm:/bin/bash
```

Step 2
After that we have removed all the unnecessary default vendor accounts into the `/etc/passwd` file from our system, we will remove all default vendor accounts into the `/etc/group` file.

- Type the following commands to delete all default users groups accounts listed below.

```
[root@deep /]# groupdel adm
[root@deep /]# groupdel lp
[root@deep /]# groupdel news
[root@deep /]# groupdel games
[root@deep /]# groupdel dip
```

Once the above list of group users has been deleted from your Linux system the `/etc/group` file will look like this:

```
root:x:0:root
bin:x:1:root,bin,daemon
daemon:x:2:root,bin,daemon
sys:x:3:root,bin
tty:x:5:
disk:x:6:root
mem:x:8:
kmem:x:9:
wheel:x:10:root
mail:x:12:mail
uucp:x:14:uucp
man:x:15:
lock:x:54:
nobody:x:99:
users:x:100:
slocate:x:21:
floppy:x:19:
utmp:x:22:
rpm:x:37:
```

Step 3
Now, you can add all the necessary and allowed users into the system. Below I show you how you should add new user into your Linux server. Adding a new user into your server mean that you have to create the username and assign him/her a password.

- To add a new user on your system, use the following command:
 `[root@deep /]# **useradd username**`

 For example:
 `[root@deep /]# **useradd admin**`

- To add or change password for user on your system, use the following command:
 `[root@deep /]# **passwd username**`

 For example:
 `[root@deep /]# **passwd admin**`

 The output should look something like this:
  ```
  Changing password for user admin
  New UNIX password:
  Retype new UNIX password:
  passwd: all authentication tokens updated successfully
  ```

Step 4
The immutable bit can be used to prevent accidentally deleting or overwriting a file that must be protected. It also prevents someone from creating a symbolic link to this file, which has been the source of attacks involving the deletion of `/etc/passwd`, `/etc/shadow`, `/etc/group` or `/etc/gshadow` files.

- To set the immutable bit on the passwords and groups files, use the following commands:
  ```
  [root@deep /]# chattr +i /etc/passwd
  [root@deep /]# chattr +i /etc/shadow
  [root@deep /]# chattr +i /etc/group
  ```

```
[root@deep /]# chattr +i /etc/gshadow
```

WARNING: In the future, if you intend to add or delete users, passwords, user groups, or group files, you must unset the immutable bit on all those files or you will not be able to make and update your changes. Also if you intend to install an `RPM` program that will automatically add a new user to the different immunized `passwd` and `group` files, then you will receive an error message during the install if you have not unset the immutable bit from those files.

- To unset the immutable bit on the passwords and groups files, use the commands:

```
[root@deep /]# chattr -i /etc/passwd
[root@deep /]# chattr -i /etc/shadow
[root@deep /]# chattr -i /etc/group
[root@deep /]# chattr -i /etc/gshadow
```

Control mounting a file system

You can have more control on mounting file systems like `/var/lib`, `/home` or `/tmp` partitions with some nifty options like **`noexec`**, **`nodev`**, and **`nosuid`**. This can be setup in the `/etc/fstab` file. The `fstab` file contains descriptive information about the various file system mount options; each line addresses one file system.

Information related to security options in the `fstab` text file are:

✓	`defaults`	Allow everything (quota, read-write, and suid) on this partition.
✓	`noquota`	Do not set users quotas on this partition.
✓	`nosuid`	Do not set SUID/SGID access on this partition.
✓	`nodev`	Do not set character or special devices access on this partition.
✓	`noexec`	Do not set execution of any binaries on this partition.
✓	`quota`	Allow users quotas on this partition.
✓	`ro`	Allow read-only on this partition.
✓	`rw`	Allow read-write on this partition.
✓	`suid`	Allow SUID/SGID access on this partition.

NOTE: For more information on options that you can set in this file (`fstab`) see the man pages about `mount (8)`.

Step 1

- Edit the **`fstab`** file (`vi /etc/fstab`) and change it depending of your needs.

For example change:

```
LABEL=/home          /home       ext3    defaults 1 2
LABEL=/tmp           /tmp        ext3    defaults 1 2
LABEL=/var/lib       /var/lib    ext3    defaults 1 2
```

To read:

```
LABEL=/home          /home       ext3    defaults,nosuid 1 2
LABEL=/tmp           /tmp        ext3    defaults,nosuid,noexec 1 2
LABEL=/var/lib       /var/lib    ext3    defaults,nodev  1 2
```

Meaning, **<nosuid>**, do not allow set-user-identifier or set-group-identifier bits to take effect, **<nodev>**, do not interpret character or block special devices on this file system partition, and **<noexec>**, do not allow execution of any binaries on the mounted file system.

Step 2
Once you have made the necessary adjustments to the `/etc/fstab` file, it is time to inform the Linux system about the modifications.

- This can be accomplished with the following commands:

```
[root@deep /]# mount /var/lib -oremount
[root@deep /]# mount /home -oremount
[root@deep /]# mount /tmp -oremount
```

Each file system that has been modified must be remounted with the command show above. In our example we have modified the `/var/lib`, `/home`, and `/tmp` file system and it is for this reason that we remount these files systems with the above commands.

- You can verify if the modifications have been correctly applied to the Linux system with the following command:

```
[root@deep /]# cat /proc/mounts
rootfs        /           rootfs  rw 0 0
/dev/root     /           ext3    rw 0 0
/proc         /proc       proc    rw 0 0
/dev/sda1     /boot       ext3    rw 0 0
/dev/sda8     /chroot     ext3    rw 0 0
none          /dev/pts    devpts  rw 0 0
/dev/sda7     /home       ext3    rw,nosuid 0 0
none          /dev/shm    tmpfs   rw 0 0
/dev/sda11    /tmp        ext3    rw,nosuid,noexec 0 0
/dev/sda6     /usr        ext3    rw 0 0
/dev/sda9     /var        ext3    rw 0 0
/dev/sda10    /var/lib    ext3    rw,nodev 0 0
```

This command will show you all the files systems on your Linux server with parameters applied to them.

Mounting the `/usr` directory of Linux as read-only

It is allowable to mount your `/usr` partition read-only since this is, by definition, static data. Of course, anyone with `root` access can remount it as writable, but a generic attack script may not know this. On many Linux variants this directory resides in its own partition and the default parameter is to mount it as read-write. We can change this parameter to make it read-only for better security. Please be sure that your `/usr` directory is on its own partition or the following hack will not work for you.

Step 1
Mounting the /usr partition as read-only eliminates possible problems that someone may try to change or modify vital files inside it. To mount the /usr file system of Linux as read-only, follow the simple steps below.

- Edit the **fstab** file (vi /etc/fstab) and change the line:

```
LABEL=/usr        /usr    ext3    defaults    1 2
```

To read:

```
LABEL=/usr        /usr    ext3    defaults,ro 1 2
```

We add the "ro" option to this line to specify to mount this partition as read-only.

Step 2
Make the Linux system aware about the modification you have made to the /etc/fstab file.

- This can be accomplished with the following command:

```
[root@deep /]# mount /usr -oremount
```

- Then test your results with the following command:

```
[root@deep /]# cat /proc/mounts
rootfs        /           rootfs  rw 0 0
/dev/root     /           ext3    rw 0 0
/proc         /proc       proc    rw 0 0
/dev/sda1     /boot       ext3    rw 0 0
/dev/sda8     /chroot     ext3    rw 0 0
none          /dev/pts    devpts  rw 0 0
/dev/sda7     /home       ext3    rw,nosuid 0 0
none          /dev/shm    tmpfs   rw 0 0
/dev/sda11    /tmp        ext3    rw,nosuid,noexec 0 0
/dev/sda6     /usr        ext3    ro 0 0
/dev/sda9     /var        ext3    rw 0 0
/dev/sda10    /var/lib    ext3    rw,nodev 0 0
```

If you see something like: /dev/sda6 /usr ext3 **ro** 0 0, congratulations!

> **WARNING:** If in the future you want to install some RPM package or program from source code, it is important to reset the modification you have made to the /usr directory to its initial state (read-write) or you will not be able to install new software because the /usr partition is set as read-only. All you have to do if you want to put the /usr partition to its original state is to edit the /etc/fstab file again and remove the "ro" option then remount the /usr file system with the "mount -oremount" command again.

Tighten scripts under `/etc/init.d`

Fix the permissions of the script files that are responsible for starting and stopping all your normal processes that need to run at boot time.

- To fix the permissions of those files, use the following command:
 `[root@deep /]# chmod 0700 /etc/init.d/*`

Which means just the super-user "`root`" is allowed to Read, Write, and Execute scripts files on this directory. I don't think regular users need to know what's inside those script files.

> **WARNING:** If you install a new program or update a program that use the init system V script located under `/etc/init.d/` directory, don't forget to change or verify the permission of this script file again.

Tighten scripts under `/etc/cron.daily/`

As for the above hack, we can tighten the security of all script files that are responsible for executing scheduled job on our server. Those files have a default permission mode of (`0755-rwxr-xr-x`), which is too high for what they should accomplish.

- To fix the permissions of those files, use the following command:
 `[root@deep /]# chmod 0550 /etc/cron.daily/*`

The same is true for other cron directories under the `/etc` directory of your system. If files exist under the other cron directories, then use the above command to change their default permission mode for better security.

> **WARNING:** If you install a new program or update a program that provides and install a cron file on your server, don't forget to change or verify the permission of this script file again.

Bits from root-owned programs

A regular user will be able to run a program as root if it is set to `SUID` root. All programs and files on your computer with the 's' bits appearing on its mode, have the `SUID` (`-rws`r-xr-x) or `SGID` (`-r-xr-s`r-x) bit enabled. Because these programs grant special privileges to the user who is executing them, it is important to remove the '`s`' bits from root-owned programs that won't absolutely require such privilege.

This can be accomplished by executing the command **`chmod a-s`** with the name(s) of the `SUID/SGID` files as its arguments.

Such programs include, but aren't limited to:

- ✓ Programs you never use.
- ✓ Programs that you don't want any non-root users to run.
- ✓ Programs you use occasionally, and don't mind having to `su` (1) to root to run.

Step 1
We've placed an asterisk (*****) next to each program we personally might disable and consider being not absolutely required for the duty work of the server. Remember that your system needs some suid root programs to work properly, so be careful.

- To find all files with the '`s`' bits from root-owned programs, use the command:

```
[root@deep]# find / -type f \( -perm -04000 -o -perm -02000 \) -exec ls -l {} \;

*-rwsr-xr-x    1 root     root        34296 Mar 27 20:40 /usr/bin/chage
*-rwsr-xr-x    1 root     root        36100 Mar 27 20:40 /usr/bin/gpasswd
-rwxr-sr-x     1 root     slocate     25020 Jun 25  2001 /usr/bin/slocate
-r-s--x--x     1 root     root        15104 Mar 13 20:44 /usr/bin/passwd
*-r-xr-sr-x    1 root     tty          6920 Mar 14 15:24 /usr/bin/wall
*-rws--x--x    1 root     root        12072 Apr  1 18:26 /usr/bin/chfn
*-rws--x--x    1 root     root        11496 Apr  1 18:26 /usr/bin/chsh
*-rws--x--x    1 root     root         4764 Apr  1 18:26 /usr/bin/newgrp
*-rwxr-sr-x    1 root     tty          8584 Apr  1 18:26 /usr/bin/write
-rwsr-xr-x     1 root     root        21080 Apr 15 00:49 /usr/bin/crontab
*-rwsr-xr-x    1 root     root        32673 Apr 18 17:40 /usr/sbin/ping6
*-rwsr-xr-x    1 root     root        13994 Apr 18 17:40 /usr/sbin/traceroute6
-rwxr-sr-x     1 root     utmp         6604 Jun 24  2001 /usr/sbin/utempter
-rws--x--x     1 root     root        22388 Apr 15 18:15 /usr/sbin/userhelper
*-rwsr-xr-x    1 root     root        17461 Apr 19 12:35 /usr/sbin/usernetctl
*-rwsr-xr-x    1 root     root        35192 Apr 18 17:40 /bin/ping
*-rwsr-xr-x    1 root     root        60104 Apr  1 18:26 /bin/mount
*-rwsr-xr-x    1 root     root        30664 Apr  1 18:26 /bin/umount
-rwsr-xr-x     1 root     root        19116 Apr  8 12:02 /bin/su
-r-sr-xr-x     1 root     root       120264 Apr  9 23:24 /sbin/pwdb_chkpwd
-r-sr-xr-x     1 root     root        16992 Apr  9 23:24 /sbin/unix_chkpwd
*-rwxr-sr-x    1 root     root        14657 Apr 19 12:35 /sbin/netreport
```

Step 2

- To disable the suid bits on selected programs above, use the following commands:

```
[root@deep /]# chmod a-s /usr/bin/chage
[root@deep /]# chmod a-s /usr/bin/gpasswd
[root@deep /]# chmod a-s /usr/bin/wall
[root@deep /]# chmod a-s /usr/bin/chfn
[root@deep /]# chmod a-s /usr/bin/chsh
[root@deep /]# chmod a-s /usr/bin/newgrp
[root@deep /]# chmod a-s /usr/bin/write
[root@deep /]# chmod a-s /usr/sbin/ping6
[root@deep /]# chmod a-s /usr/sbin/traceroute6
[root@deep /]# chmod a-s /usr/sbin/usernetctl
[root@deep /]# chmod a-s /bin/ping
[root@deep /]# chmod a-s /bin/mount
[root@deep /]# chmod a-s /bin/umount
[root@deep /]# chmod a-s /sbin/netreport
```

Don't let internal machines tell the server what their MAC address is

To avoid the risk that a user could easily change a computers IP address and appear as someone else to the firewall, you can force the ARP cache entries of Linux using the arp command utility. A special option can be used with the arp utility to avoid letting INTERNAL machines tell the server what their MAC (**M**edia **A**ccess **C**ontrol) address is and the IP address associated with it.

Step1
ARP is a small utility, which manipulates the kernel's ARP (**A**ddress **R**esolution **P**rotocol) cache. Through all possible options associated with this utility, the primary one is clearing an address mapping entry and manually setting up one. In the hope to more secure our server from the INTERNAL, we will manually set MAC address (sometimes called Hardware addresses) of all known computers in our network statically by using static ARP entries.

- For each IP address of INTERNAL computers in your network, use the following command to know the MAC address associate with the IP address:

```
[root@deep /]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:50:DA:C6:D3:FF
          inet addr:207.35.78.3  Bcast:207.35.78.32  Mask:255.255.255.224
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1887318 errors:0 dropped:0 overruns:1 frame:0
          TX packets:2709329 errors:0 dropped:0 overruns:0 carrier:1
          collisions:18685 txqueuelen:100
          Interrupt:10 Base address:0xb000

eth1      Link encap:Ethernet  HWaddr 00:50:DA:C6:D3:09
          inet addr:192.168.1.11  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:182937 errors:0 dropped:0 overruns:0 frame:0
          TX packets:179612 errors:0 dropped:0 overruns:0 carrier:0
          collisions:7434 txqueuelen:100
          Interrupt:11 Base address:0xa800

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:3924  Metric:1
          RX packets:7465 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7465 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
```

The MAC (**M**edia **A**ccess **C**ontrol) address will be the letters and numbers that come after "**HWaddr**" (the Hardware Address). In the above example our MAC address are:

00:50:DA:C6:D3:FF for the interface eth0 and **00:50:DA:C6:D3:09** for the interface eth1.

Step 2
Once we know the MAC (**M**edia **A**ccess **C**ontrol) address associated with IP address, we can add them manually to the ARP entries of the Linux server.

- To add manually MAC address to ARP entries, use the following commands:

```
[root@deep /]# arp -s 207.35.78.3 00:50:DA:C6:D3:FF
[root@deep /]# arp -s 192.168.1.11 00:50:DA:C6:D3:09
```

The "**-s**" option means to manually create an ARP address mapping entry for host hostname with hardware address set to hw_addr class. You can add you ARP commands to the /etc/rc.local file if you want to keep your configuration if the system reboots.

Step 3

- To verify if the modifications have been added to the system, use the following command:

```
[root@deep /]# arp
Address       Hwtype  Hwaddress           Flags Mask  Iface
207.35.78.3   ether   00:20:78:13:86:92   CM          eth1
192.168.1.11  ether   00:E0:18:90:1B:56   CM          eth1
```

WARNING: If you receive error message like: **SIOCSARP: Invalid argument**, it is because the MAC (Media Access Control) address you want to add is the one of your server. You must add only MAC address of INTERNAL computers in your private network. This hack doesn't apply to external node on the Internet.

You can now be reassured that someone will not change the system's IP address of an INTERNAL system and get through. If they do change the IP address, the server simply won't talk to them. With the new iptables tool of Linux, which replace the old ipchains utility for packet filter administration and firewall setup, MAC addresses can be filtered and configured in the firewall rules too.

Unusual or hidden files

It is important to look everywhere on the system for unusual or hidden files (files that start with a period and are normally not shown by the "**ls**" command), as these can be used to hide tools and information (password cracking programs, password files from other systems, etc.). A common technique on UNIX systems is to put a hidden directory or file in a user's account with an unusual name, something like '. . .' or '. . ' (dot dot space) or '. . ^G' (dot dot control-G). The find program can be used to look for hidden files.

- To look for hidden files, use the following commands:

```
[root@deep /]# find / -name ".. " -print -xdev

[root@deep /]# find / -name ".*" -print -xdev | cat –v
/etc/skel/.bash_logout
/etc/skel/.bash_profile
/etc/skel/.bashrc
/etc/.pwd.lock
/root/.bash_logout
/root/.Xresources
/root/.bash_profile
/root/.bashrc
/root/.cshrc
/root/.tcshrc
/root/.bash_history
/usr/lib/perl5/5.6.1/i386-linux/.packlist
/home/admin/.bash_logout
/home/admin/.bash_profile
/home/admin/.bashrc
/.autofsck
```

Finding Group and World Writable files and directories

Group and world writable files and directories, particularly system files (partitions), can be a security hole if a cracker gains access to your system and modifies them. Additionally, world-writable directories are dangerous, since they allow a cracker to add or delete files as he or she wishes in these directories. In the normal course of operation, several files will be writable, including some from the `/dev`, `/var/mail` directories, and all symbolic links on your system.

- To locate all group & world-writable files on your system, use the command:

```
[root@deep /]# find / -type f \( -perm -2 -o -perm -20 \) -exec ls -lg {} \;
-rw-rw-r--    1 root     utmp       107904 Jun 17 12:04 /var/log/wtmp
-rw-rw-r--    1 root     utmp         4608 Jun 17 12:04 /var/run/utmp
```

- To locate all group & world-writable directories on your system, use the command:

```
[root@deep /]# find / -type d \( -perm -2 -o -perm -20 \) -exec ls -ldg {} \;
drwxrwxr-x   12 root     man          4096 Jun 17 06:50 /var/cache/man/X11R6
drwxrwxr-x    2 root     man          4096 Mar 25 09:17 /var/cache/man/X11R6/cat1
drwxrwxr-x    2 root     man          4096 Mar 25 09:17 /var/cache/man/X11R6/cat2
drwxrwxr-x    2 root     man          4096 Mar 25 09:17 /var/cache/man/X11R6/cat3
drwxrwxr-x    2 root     man          4096 Mar 25 09:17 /var/cache/man/X11R6/cat4
drwxrwxr-x    2 root     man          4096 Mar 25 09:17 /var/cache/man/X11R6/cat5
drwxrwxr-x    2 root     man          4096 Mar 25 09:17 /var/cache/man/X11R6/cat6
drwxrwxr-x    2 root     man          4096 Mar 25 09:17 /var/cache/man/X11R6/cat7
drwxrwxr-x    2 root     man          4096 Mar 25 09:17 /var/cache/man/X11R6/cat8
drwxrwxr-x    2 root     man          4096 Mar 25 09:17 /var/cache/man/X11R6/cat9
drwxrwxr-x    2 root     man          4096 Mar 25 09:17 /var/cache/man/X11R6/catn
drwxrwxr-x    2 root     man          4096 Mar 25 09:17 /var/cache/man/cat1
drwxrwxr-x    2 root     man          4096 Mar 25 09:17 /var/cache/man/cat2
drwxrwxr-x    2 root     man          4096 Mar 25 09:17 /var/cache/man/cat3
drwxrwxr-x    2 root     man          4096 Mar 25 09:17 /var/cache/man/cat4
drwxrwxr-x    2 root     man          4096 Mar 25 09:17 /var/cache/man/cat5
drwxrwxr-x    2 root     man          4096 Mar 25 09:17 /var/cache/man/cat6
drwxrwxr-x    2 root     man          4096 Mar 25 09:17 /var/cache/man/cat7
drwxrwxr-x    2 root     man          4096 Mar 25 09:17 /var/cache/man/cat8
drwxrwxr-x    2 root     man          4096 Mar 25 09:17 /var/cache/man/cat9
drwxrwxr-x    2 root     man          4096 Mar 25 09:17 /var/cache/man/catn
drwxrwxr-x   12 root     man          4096 Jun 17 06:50 /var/cache/man/local
drwxrwxr-x    2 root     man          4096 Mar 25 09:17 /var/cache/man/local/cat1
drwxrwxr-x    2 root     man          4096 Mar 25 09:17 /var/cache/man/local/cat2
drwxrwxr-x    2 root     man          4096 Mar 25 09:17 /var/cache/man/local/cat3
drwxrwxr-x    2 root     man          4096 Mar 25 09:17 /var/cache/man/local/cat4
drwxrwxr-x    2 root     man          4096 Mar 25 09:17 /var/cache/man/local/cat5
drwxrwxr-x    2 root     man          4096 Mar 25 09:17 /var/cache/man/local/cat6
drwxrwxr-x    2 root     man          4096 Mar 25 09:17 /var/cache/man/local/cat7
drwxrwxr-x    2 root     man          4096 Mar 25 09:17 /var/cache/man/local/cat8
drwxrwxr-x    2 root     man          4096 Mar 25 09:17 /var/cache/man/local/cat9
drwxrwxr-x    2 root     man          4096 Mar 25 09:17 /var/cache/man/local/catn
drwxrwxr-x    3 root     lock         4096 Jun 17 06:49 /var/lock
drwxrwxr-x    2 root     root         4096 Apr 19 12:35 /var/run/netreport
drwxrwxr-x    2 root     12           4096 Jun 17 12:30 /var/spool/mail
drwxrwxrwt    2 root     root         4096 Jun 17 11:29 /var/tmp
drwxrwxrwt    2 root     root         4096 Jun 17 06:52 /tmp
```

WARNING: A file and directory integrity checker like "**`Tripwire`**" software can be used regularly to scan, manage and find modified group or world writable files and directories easily. See later in this book for more information about `Tripwire`.

Unowned files

Don't permit any unowned file on your server. Unowned files may also be an indication that an intruder has accessed your system. If you find unowned file or directory on your system, verify its integrity, and if all looks fine, give it an owner name. Some time you may uninstall a program and get an unowned file or directory related to this software; in this case you can remove the file or directory safely.

- To locate files on your system that do not have an owner, use the following command:
 `[root@deep /]# find / -nouser -o -nogroup`

WARNING: It is important to note that files reported under `/dev/` directory don't count.

Finding `.rhosts` files

Finding all existing `.rhosts` files that could exist on your server should be a part of your regular system administration duties, as these files should not be permitted on your system. Remember that a cracker only needs one insecure account to potentially gain access to your entire network.

Step 1

If you are doing a new install of Linux (like we did) you should not have any `.rhosts` files on your system. If the result returns nothing, then you are safe and your system contain no `.rhosts` files in the `/home` directory at this time.

- You can locate all existing `.rhosts` files on your system with the following command:
 `[root@deep /]# find /home -name .rhosts`

Step 2

You can also use a `cron` job to periodically check for, report the contents of, and delete `$HOME/.rhosts` files. Also, users should be made aware that you regularly perform this type of audit, as directed by your security policy.

- Create the **`rhosts.cron`** file (`touch /etc/cron.daily/rhosts.cron`) and add the following lines inside the script file.

```
#!/bin/sh

/usr/bin/find /home -name .rhosts | (cat <<EOF
This is an automated report of possible existent ..rhosts files on
the server deep.openna.com, generated by the find utility command.

New detected ..rhosts. files under the ./home/. directory include:
EOF
cat
) | /bin/mail -s "Content of .rhosts file audit report" root
```

- Now make this script executable, verify the owner, and change the group to "`root`".
 `[root@deep /]# chmod 550 /etc/cron.daily/rhosts.cron`
 `[root@deep /]# chown 0.0 /etc/cron.daily/rhosts.cron`

Each day mail will be sent to "`root`" with a subject:" Content of .rhosts file audit report" containing potential new `.rhosts` files.

Physical hard copies of all-important logs

One of the most important security considerations is the integrity of the different log files under the `/var/log` directory on your server. If despite each of the security functions put in place on our server, a cracker can gain access to it; our last defence is the log file system, so it is very important to consider a method of being sure of the integrity of our log files.

If you have a printer installed on your server, or on a machine on your network, a good idea would be to have actual physical hard copies of all-important logs. This can be easily accomplished by using a continuous feed printer and having the `syslog` program sending all logs you think are important out to `/dev/lp0` (the printer device). Cracker can change the files, programs, etc on your server, but can do nothing when you have a printer that prints a real paper copy of all of your important logs.

As an example:
For logging of all `telnet`, `mail`, `boot` messages and `ssh` connections from your server to the printer attached to THIS server, you would want to add the following line to the `/etc/syslog.conf` file:

Step 1

- Edit the **`syslog.conf`** file (`vi /etc/syslog.conf`) and add at the end of this file the following line:

```
authpriv.*;mail.*;local7.*;auth.*;daemon.info /dev/lp0
```

Step 2

- Now restart your `syslog` daemon for the change to take effect:

```
[root@deep /]# /etc/init.d/syslog restart
Shutting down kernel logger:          [OK]
Shutting down system logger:          [OK]
Starting system logger:               [OK]
Starting kernel logger:               [OK]
```

As an example:
For logging of all `telnet`, `mail`, `boot` messages and `ssh` connections from your server to the printer attached to a REMOTE server in your local network, then you would want to add the following line to `/etc/syslog.conf` file on the REMOTE server.

Step 1

- Edit the **`syslog.conf`** file (`vi /etc/syslog.conf`) on the REMOTE server (for example: `printer.openna.com`) and add at the end of this file the following line:

```
authpriv.*;mail.*;local7.*;auth.*;daemon.info /dev/lp0
```

If you don't have a printer in your network, you can also copy all the log files to another machine; simply omit the above first step of adding `/dev/lp0` to your `syslog.conf` file on remote and go directly to the "**-r**" option second step on remote. Using the feature of copying all the log files to another machine will give you the possibility to control all `syslog` messages on one host and will tear down administration needs.

Step 2
Since the default configuration of the `syslog` daemon is to not receive any messages from the network, we must enable on the REMOTE server the facility to receive messages from the network. To enable the facility to receive messages from the network on the REMOTE server, add the following option "**`-r`**" to your `syslog` daemon script file (only on the REMOTE host):

- Edit the `syslog` daemon (`vi +24 /etc/rc.d/init.d/syslog`) and change:

  ```
  daemon syslogd -m 0
  ```

 To read:

  ```
  daemon syslogd -r -m 0
  ```

Step 3

- Restart your syslog daemon on the remote host for the change to take effect:

  ```
  [root@mail /]# /etc/init.d/syslog restart
  Shutting down kernel logger:          [OK]
  Shutting down system logger:          [OK]
  Starting system logger:               [OK]
  Starting kernel logger:               [OK]
  ```

Step 4

- Edit the **`syslog.conf`** file (`vi /etc/syslog.conf`) on the LOCAL server, and add at the end of this file the following line:

  ```
  authpriv.*;mail.*;local7.*;auth.*;daemon.info @printer
  ```

Where (`printer`) represent the hostname of the REMOTE server. Now if anyone ever hacks your machine and attempts to erase vital system logs, you still have a hard copy of everything. It should then be fairly simple to trace where they came from and deal with it accordingly.

Step 5

- Restart your `syslog` daemon on the LOCAL server for the change to take effect:

  ```
  [root@deep /]# /etc/init.d/syslog restart
  Shutting down kernel logger:          [OK]
  Shutting down system logger:          [OK]
  Starting system logger:               [OK]
  Starting kernel logger:               [OK]
  ```

WARNING: Never use your Gateway Server as a host to control all `syslog` messages; this is a very bad idea. More options and strategies exist with the `sysklogd` program, see the man pages about `sysklogd (8), syslog(2),` and `syslog.conf(5)` for more information.

Getting some more security by removing manual pages

Here we have to think a little bit about manual pages installed on all Linux system. Manual pages also known as man-pages are compressed files located under the `/usr/share/man` directory on your system. These documentation files are very useful to get quick information on how service, program, commands, and configuration files of specific software work. These files are readable by the `man` program and depend of other installed software on Linux to work and display the information.

On production servers where specific task are assigned and where we only run services to the internal or external, does we really need to have these manual pages and related software installed? Do we will connect to these production servers to read these manual pages? Does this is really important to have them duplicated on all of our different servers? Personally, I don't think because we can have all of these useful documentation files available on our Linux workstation or development server each time we need to consult them.

If you have made attention to what we have done previously to secure our server, you will remember that most of all group and world-writable directories on our system comes from the `/var/cache` directory which is owned by the `man` program associated with manual pages. By removing manual pages and related software from our server, we can get some more security and save some not negligible space which could help when we scan our server with integrity tool like `Tripwire`. This also allow us to remove other software directly related to `man` program and limit the number of installed component on our production server without scarifying in the functionally of the server. If this is what you want to do, here are the steps to follow.

Step 1
First of all, we should remove the `man` software from our system. The `man` software is the program we use to read manual pages. By removing this software we eliminate most of all group and world-writable directories from our system.

- To remove the `man` software, use the following command:
```
[root@deep /]# rpm -e man
```

Step 2
Once the above software has been removed, we can continue with `groff`. `Groff` is a document formatting system that takes standard text and formatting commands as input and produces formatted output. This software is used by `man` to format man-pages.

- To remove the `groff` software, use the following command:
```
[root@deep /]# rpm -e groff
```

Step 3
Because we don't use manual pages anymore on our production servers, we can remove all man-pages that are already installed and available under the `/usr/share/man` directory.

- To remove all preinstalled man-pages from your server, use the following commands:
```
[root@deep /]# cd /usr/share/man/
[root@deep man]# rm -f man*/*.gz
```

Step 4
Finally, it is important to note that any future installation and upgrade of RPM packages on the system should be made with the "--excludedocs" option. This RPM option allow us to install or upgrade the RPM package without the need to install the documentation part that may comes with the software. For example, if I want to install or upgrade the bind package, I will use the following RPM command.

- To install or upgrade RPM without documentation, use the following command:
 `[root@deep /]# rpm –Uvh --exculdedocs bind-version.i386.rpm`

System is compromised!

If you believe that your system has been compromised, contact CERT ® Coordination Center or your representative in FIRST (Forum of Incident Response and Security Teams).

Internet Email: cert@cert.org
CERT Hotline: (+1) 412-268-7090
Facsimile: (+1) 412-268-6989
CERT/CC personnel answer 8:00 a.m. – 8:00 p.m. EST (GMT –5)/EDT (GMT –4)) on working days; they are on call for emergencies during other hours and on weekends and holidays.

CHAPTER 4

Pluggable Authentication Modules

IN THIS CHAPTER

1. **The password length**
2. **Disabling console program access**
3. **Disabling all console access**
4. **The Login access control table**
5. **Tighten console permissions for privileged users**
6. **Putting limits on resource**
7. **Controlling access time to services**
8. **Blocking; `su` to `root`, by one and sundry**
9. **Using `sudo` instead of `su` for logging as super-user**

Linux PAM

Abstract
The **P**luggable **A**uthentication **M**odules (PAM) consists of shared libraries, which enable administrators to choose how applications authenticate users.

Basically, PAM enables the separation of authentication schemes from the applications. This is accomplished by providing a library of functions that applications can use for requesting user authentications. ssh, pop, imap, etc. are PAM-aware applications, hence these applications can be changed from providing a password to providing a voice sample or fingerprint by simply changing the PAM modules without having to rewrite any code in these applications.

The configuration files of the PAM modules are located in the directory /etc/pam.d and the modules (shared libraries) themselves are located in the directory /lib/security. The /etc/pam.d directory has a collection of named files of its own, e.g. ssh, pop, imap, etc. PAM-aware applications that do not have a configuration file will automatically be pointed to the default configuration file 'other'.

In the next section we will set up some recommended minimum-security restrictions using PAM.

The password length
The minimum acceptable password length by default when you install your Linux system is five. This means that when a new user is given access to the server, his/her password length will be at minimum five mixes of character strings, letter, number, special character etc. This is not enough and must be eight or more. The password length under Linux by the use of its PAM feature is controlled by five arguments: minlen, dcredit, ucredit, lcredit, and ocredit.

Step 1
To prevent non-security-minded people or administrators from being able to enter just five characters for the valuable password, edit the rather important /etc/pam.d/system-auth file and enforce the minimum password length.

- Edit the **system-auth** file (vi /etc/pam.d/system-auth) and change the line:

  ```
  password  required  /lib/security/pam_cracklib.so retry=3 type=
  ```

 To read:

  ```
  password  required  /lib/security/pam_cracklib.so retry=3 minlen=12 type=
  ```

After changing the above line, the /etc/pam.d/system-auth file should look like this:

```
#%PAM-1.0
auth       required     /lib/security/pam_env.so
auth       sufficient   /lib/security/pam_unix.so likeauth nullok
auth       required     /lib/security/pam_deny.so
account    required     /lib/security/pam_unix.so
password   required     /lib/security/pam_cracklib.so retry=3 minlen=12 type=
password   sufficient   /lib/security/pam_unix.so nullok use_authtok md5 shadow
password   required     /lib/security/pam_deny.so
session    required     /lib/security/pam_limits.so
session    required     /lib/security/pam_unix.so
```

WARNING: It is important to note that when you set the password for a user under '`root`' account, then these restrictions don't apply!! This is the case on all Unix OS. The super-user '`root`' can override pretty much everything. Instead, log as the user account from which you apply this restriction and try to change the password. You will see that it works.

You need to keep in mind that this module includes a credit mechanism. E.g. if you define **`minlen=12`**, then you will get 1 credit for e.g. including a single digit number in your password, or for including a non-alphanumeric character. Getting 1 credit means that the module will accept a password of the length of minlen-credit. When you check the parameters of the cracklib module, you will see that it has some parameters that let you define what a credit is (http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/pam.html).

For example:

minlen	The following password was accepted
14	gjtodgsdf1$

You can see that I got 1 credit for a alphanumeric character and a credit for each non-alphanumeric character. "`gjtodgsdf1`$" has a length of 11, 1 credit for alpha-numeric, 2 credits for non-alphanumeric character (1 and $) which gives me a credit of 3, hence the password length of 11 was accepted.

At any rate, the minimum length is adjusted by the mixture of types of characters used in the password. Using digits (up to the number specified with the "`dcredit=`" parameter, which defaults to 1) or uppercase letters "`ucredit`" or lowercase letters "`lcredit`" or other types of letters "`ocredit`" will decrease the minimum length by up to four since the default parameter for these arguments is 1 and there is four different arguments that you can add.

A password with 9 lowercase letters in it will pass a minimum length set to 10 unless "`lcredit=0`" is used, because a credit is granted for the use of a lowercase letter. If the mixture includes an uppercase letter, a lowercase letter, and a digit, then a minlength of 8 effectively becomes 5.

NOTE: With the new `MD5` passwords capability, which is installed by default in all modern Linux operating system, a long password can be used now (up to 256 characters), instead of the Unix standard eight letters or less. If you want to change the password length of 8 characters to example 16 characters, all you have to do is to replace the number 12 by 20 in the "`minlen=12`" line of the `/etc/pam.d/system-auth` file.

Disabling console program access

In a safe environment, where we are sure that console is secured because passwords for BIOS and GRUB or LILO are set and all physical power and reset switches on the system are disabled, it may be advantageous to entirely disable all console-equivalent access to programs like poweroff, reboot, and halt for regular users on your server.

- To do this, run the following command:
  ```
  [root@deep /]# rm -f /etc/security/console.apps/<servicename>
  ```

Where <servicename> is the name of the program to which you wish to disable console-equivalent access.

- To disable console program access, use the following commands:
  ```
  [root@deep /]# rm -f /etc/security/console.apps/halt
  [root@deep /]# rm -f /etc/security/console.apps/poweroff
  [root@deep /]# rm -f /etc/security/console.apps/reboot
  ```

This will disable console-equivalent access to programs halt, poweroff, and reboot.

Disabling all console access

The Linux-PAM library installed by default on your system allows the system administrator to choose how applications authenticate users, such as for console access, program and file access.

Step 1
In order to disable all these accesses for the users, you must comment out all lines that refer to **pam_console.so** in the /etc/pam.d directory. This step is a continuation of the hack "Disabling console program access". The following script will do the trick automatically for you.

- As 'root' creates the **disabling.sh** script file (touch disabling.sh) and add the following lines inside:

  ```
  # !/bin/sh
  cd /etc/pam.d
  for i in * ; do
  sed '/[^#].*pam_console.so/s/^/#/' < $i > foo && mv foo $i
  done
  ```

Step 2
Now, we have to make the script executable and run it.

- This can be done with the following commands:
  ```
  [root@deep /]# chmod 700 disabling.sh
  [root@deep /]# ./disabling.sh
  ```

This will comment out all lines that refer to **pam_console.so** for all files located under /etc/pam.d directory. Once the script has been executed, you can remove it from your system.

The Login access control table

On a server environment where authorized and legitimate logins can come from everywhere, it is important to have the possibility to use a security file which allows us to have more control over users who can connect to the server. What we are looking here is to have more control on not allowing some legitimated accounts to login from anywhere. Fortunately, this file exists and is called "`access.conf`", you can find it under your `/etc/security` directory.

The `access.conf` file which comes already installed with your native Linux system allow us to control which authorized users can/cannot log in to the server or to the console and from where. Don't forget that users access can come everywhere from remote host or directly from the console of the system. Configuration of the `access.conf` file of Linux is not complicated to understand. Below I show you how to configure it to be very restrictive and secure.

Step 1
By default denying access to every one, is the first step of a reliable security policy. In this way we eliminate the possibility of forgetting someone or to making a mistake.

- Edit the **`access.conf`** file (`vi /etc/security/access.conf`) and add the following line at the end of the file.

```
-:ALL EXCEPT root gmourani:ALL
```

This access policy means to disallow console logins as well as remote accounts login to all from anywhere except for user '`root`' and '`gmourani`'. With this choice of policy, we deny non-networked and remote logins to every user with a shell account on the system from everywhere and allow only the selected users.

Take a note that many possibilities exist as for example allowing the same users '`root`' and '`gmourani`' to log only to the system from remote host with `IP` address `207.35.78.2`. To enable this policy, all we need to do is to change the above policy to this one:

- Edit the **`access.conf`** file (`vi /etc/security/access.conf`) and add the following lines at the end of the file.

```
-:ALL EXCEPT root gmourani:207.35.78.2
-:ALL:LOCAL
```

Here the second policy line means to disallow all local access to the console for every users even for the super-user '`root`', therefore if you want to log as '`root`' you need first to log as user '`gmourani`' from remote host with `IP` address `207.35.78.2` and `su` to '`root`' (this is why I added '`root`' to the users allowed to connect from remote host `207.35.78.2`).

Step 2
To be able to use the `access.conf` feature of Linux, make sure to add the following line to `/etc/pam.d/system-auth` and `sshd` if you use this service or it will not work.

- Edit the **`login`** file (`vi /etc/pam.d/system-auth`) and add the following line.

```
account    required     /lib/security/pam_access.so
```

After adding the above line, the `/etc/pam.d/system-auth` file should look like this:

```
#%PAM-1.0
auth       required     /lib/security/pam_env.so
auth       sufficient   /lib/security/pam_unix.so likeauth nullok
auth       required     /lib/security/pam_deny.so
account    required     /lib/security/pam_unix.so
account    required     /lib/security/pam_access.so
password   required     /lib/security/pam_cracklib.so retry=3 minlen=12 type=
password   sufficient   /lib/security/pam_unix.so nullok use_authtok md5 shadow
password   required     /lib/security/pam_deny.so
session    required     /lib/security/pam_limits.so
session    required     /lib/security/pam_unix.so
```

NOTE: Please read information about possible configurations of this file inside the `access.conf` file since your policies will certainly differ from the example that I show you above.

Tighten console permissions for privileged users

The `console.perms` security file of Linux, which use the `pam_console.so` module to operate, is designed to give to privileged users at the physical console (virtual terminals and local xdm-managed X sessions) capabilities that they would not otherwise have, and to take those capabilities away when they are no longer logged in at the console.

It provides two main kinds of capabilities: file permissions and authentication. When a user logs in at the console and **no other user is currently logged in at the console**, the `pam_console.so` module will change permissions and ownership of files as described in the file `/etc/security/console.perms`.

Please note that privileged users are nothing in common with regular users you may add to the server, they are special users like `floppy`, `cdrom`, `scanner`, etc which in an networking server environment are also considered and treated as users.

Step 1
The default `console.perms` configuration file of Linux is secure enough for regular use of the system where an Xwindow interface is considered to be installed but in a highly secure environment where the **G**raphical **U**ser **I**nterface (`GUI`) is not installed or where some special devices like `sound`, `jaz`, etc have no reason to exist, we can tighten the `console.perms` security file of Linux to be more secure by eliminating non-existent or unneeded privileged users to have capabilities that they would not otherwise have.

- Edit the **`console.perms`** file (`vi /etc/security/console.perms`), and change the default lines inside this file:

```
# file classes -- these are regular expressions
<console>=tty[0-9][0-9]* vc/[0-9][0-9]* :[0-9]\.[0-9] :[0-9]
<xconsole>=:[0-9]\.[0-9] :[0-9]

# device classes -- these are shell-style globs
<floppy>=/dev/fd[0-1]* \
         /dev/floppy/* /mnt/floppy*
<sound>=/dev/dsp* /dev/audio* /dev/midi* \
        /dev/mixer* /dev/sequencer \
        /dev/sound/* /dev/beep
<cdrom>=/dev/cdrom* /dev/cdroms/* /dev/cdwriter* /mnt/cdrom*
<pilot>=/dev/pilot
<jaz>=/mnt/jaz*
<zip>=/mnt/pocketzip* /mnt/zip*
<ls120>=/dev/ls120 /mnt/ls120*
<scanner>=/dev/scanner /dev/usb/scanner*
<rio500>=/dev/usb/rio500
<camera>=/mnt/camera* /dev/usb/dc2xx* /dev/usb/mdc800*
<memstick>=/mnt/memstick*
<flash>=/mnt/flash*
<diskonkey>=/mnt/diskonkey*
<rem_ide>=/mnt/microdrive*
<fb>=/dev/fb /dev/fb[0-9]* \
     /dev/fb/*
<kbd>=/dev/kbd
<joystick>=/dev/js[0-9]*
<v4l>=/dev/video* /dev/radio* /dev/winradio* /dev/vtx* /dev/vbi* \
      /dev/video/*
<gpm>=/dev/gpmctl
<dri>=/dev/nvidia* /dev/3dfx*
<mainboard>=/dev/apm_bios

# permission definitions
<console>  0660 <floppy>     0660 root.floppy
<console>  0600 <sound>      0600 root
<console>  0600 <cdrom>      0660 root.disk
<console>  0600 <pilot>      0660 root.uucp
<console>  0600 <jaz>        0660 root.disk
<console>  0600 <zip>        0660 root.disk
<console>  0600 <ls120>      0660 root.disk
<console>  0600 <scanner>    0600 root
<console>  0600 <camera>     0600 root
<console>  0600 <memstick>   0600 root
<console>  0600 <flash>      0600 root
<console>  0600 <diskonkey>  0660 root.disk
<console>  0600 <rem_ide>    0660 root.disk
<console>  0600 <fb>         0600 root
<console>  0600 <kbd>        0600 root
<console>  0600 <joystick>   0600 root
```

```
<console>  0600 <v4l>        0600 root
<console>  0700 <gpm>        0700 root
<console>  0600 <mainboard>  0600 root
<console>  0600 <rio500>     0600 root

<xconsole> 0600 /dev/console 0600 root.root
<xconsole> 0600 <dri>        0600 root
```

To read :

```
# file classes -- these are regular expressions
<console>=tty[0-9][0-9]* vc/[0-9][0-9]* :[0-9]\.[0-9] :[0-9]

# device classes -- these are shell-style globs
<floppy>=/dev/fd[0-1]* \
         /dev/floppy/* /mnt/floppy*
<cdrom>=/dev/cdrom* /dev/cdroms/* /dev/cdwriter* /mnt/cdrom*
<pilot>=/dev/pilot
<fb>=/dev/fb /dev/fb[0-9]* \
     /dev/fb/*
<kbd>=/dev/kbd
<gpm>=/dev/gpmctl
<mainboard>=/dev/apm_bios

# permission definitions
<console>  0660 <floppy>     0660 root.floppy
<console>  0600 <cdrom>      0660 root.disk
<console>  0600 <pilot>      0660 root.uucp
<console>  0600 <fb>         0600 root
<console>  0600 <kbd>        0600 root
<console>  0700 <gpm>        0700 root
<console>  0600 <mainboard>  0600 root
```

Here we removed every privileged user related to the **G**raphical **U**ser **I**nterface and others related to `sound`, `zip` drive, `jaz` drive, `scanner`, `joystick` and `video` media at the physical console on the server.

Putting limits on resource

The `limits.conf` file located under the `/etc/security` directory can be used to control and limit resources for the users on your system. It is important to set resource limits on all your users so they can't perform **D**enial **o**f **S**ervice attacks (number of processes, amount of memory, etc) on the server. These limits will have to be set up for the user when he or she logs in.

Step 1
For example, limits for all users on your system might look like this:

- Edit the **`limits.conf`** file (`vi /etc/security/limits.conf`) and change the lines:

```
#*           soft    core        0
#*           hard    rss     10000
#@student    hard    nproc   20
```

To read:

```
*            hard    core    0
*            hard    rss     5000
*            hard    nproc   35
```

This says to prohibit the creation of core files "**core 0**", restrict the number of processes to 35 "**nproc 35**", and restrict memory usage to 5M "**rss 5000**" for everyone except the super user "root". All of the above only concerns users who have entered through the login prompt on your system. With this kind of quota, you have more control on the processes, core files, and memory usage that users may have on your system. The asterisk "*****" mean: all users that logs in on the server.

Putting an asterisk "*****" to cover all users can pose problem with daemon users account like "www" for a web server, "mysql" for a SQL database server, etc. If we put an asterisk, then, these users will be affected by the restriction and limitation of processes or memory usage.

To solve the problem, we can choose an existing group name in our system and add every regular user to this group. In this manner, the restrictions and limitations will apply to all users who are members of this group name only. A special group account named "users" can be used for this purpose. This is the recommended method on putting limit on user resources.

- Edit the **limits.conf** file (vi /etc/security/limits.conf) and change the lines:

```
#*           soft    core        0
#*           hard    rss     10000
#@student    hard    nproc   20
```

To read:

```
@users       hard    core    0
@users       hard    rss     5000
@users       hard    nproc   35
```

If you decide to use a group name like "**@users**" to control and limit resources for the users on your system, then it is important to not forget to change the GUI (**G**roup **U**ser **I**D) of these users to be "100". "100" is the numeric value of the user's ID "users".

- The command to create a new user with group name which is set by default to users is:
 `[root@deep /]# useradd -g100 admin`

The "-g100" option represents the number of the user's initial login group and in our case "100" is the group account name "users". The "admin" parameter is the user name we want to add to the group name "users".

> **WARNING:** Use the same command above for all users on your system you want to be member of the "users" group account. It is also preferable to set this parameter first before adding users to the system.

Controlling access time to services

As the Linux-PAM system said, running a well-regulated system occasionally involves restricting access to certain services in a selective manner. The time.conf security file, which is provided by the pam_time.so module of Linux, offers some time control for access to services offered by a system. Its actions are determined through the configuration file called time.conf and located under /etc/security directory.

Step 1
The time.conf file can be configured to deny access to (individual) users based on their name, the time of day, the day of week, the service they are applying for and their terminal from which they are making their request.

- Edit the **time.conf** file (vi /etc/security/time.conf), and add the following line:

 login ; tty* & !ttyp* ; !root & gmourani ; !Al0000-2400

The above time control access line means to deny all user access to console-login at all times except for the super-user 'root' and the user 'gmourani'.

Take a note that many combinations exist as described in the time.conf file, we can, for example, allow user 'admin' to access the console-login any time except at the weekend and on Tuesday from 8AM to 6PM with the following statement.

- Edit the time.conf file (vi /etc/security/time.conf), and add the following line:

 login ; * ; !admin ; !Wd0000-2400 & Tu0800-1800

Step 2
To be able to use the time.conf feature of Linux, make sure to add the following line to /etc/pam.d/system-auth and sshd if you use this service or nothing will work.

- Edit the **system-auth** file (vi /etc/pam.d/system-auth) and add the line.

 account required /lib/security/pam_time.so

After adding the line above, the /etc/pam.d/system-auth file should look like this:

```
#%PAM-1.0
auth       required     /lib/security/pam_env.so
auth       sufficient   /lib/security/pam_unix.so likeauth nullok
auth       required     /lib/security/pam_deny.so
account    required     /lib/security/pam_unix.so
account    required     /lib/security/pam_access.so
account    required     /lib/security/pam_time.so
password   required     /lib/security/pam_cracklib.so retry=3 minlen=12 type=
password   sufficient   /lib/security/pam_unix.so nullok use_authtok md5 shadow
password   required     /lib/security/pam_deny.so

session    required     /lib/security/pam_limits.so
session    required     /lib/security/pam_unix.so
```

Blocking; `su` to root, by one and sundry

The `su` (**S**ubstitute **U**ser) command allows you to become other existing users on the system. For example you can temporarily become '`root`' and execute commands as the super-user '`root`'.

Step 1
If you don't want anyone to `su` to root or want to restrict the `su` command to certain users then uncomment the following line of your `su` configuration file in the `/etc/pam.d` directory. We highly recommend that you limit the persons allowed to `su` to the `root` account.

- Edit the **`su`** file (`vi /etc/pam.d/su`) and uncomment the following line in the file:

 `auth required /lib/security/pam_wheel.so use_uid`

After this line has been uncommented, the `/etc/pam.d/su` file should look like this:

```
#%PAM-1.0
auth       sufficient   /lib/security/pam_rootok.so
auth       required     /lib/security/pam_wheel.so use_uid
auth       required     /lib/security/pam_stack.so service=system-auth
account    required     /lib/security/pam_stack.so service=system-auth
password   required     /lib/security/pam_stack.so service=system-auth
session    required     /lib/security/pam_stack.so service=system-auth
session    optional     /lib/security/pam_xauth.so
```

Which means only those who are members of the "`wheel`" group can `su` to `root`; it also includes logging. Note that the "`wheel`" group is a special account on your system that can be used for this purpose. You cannot use any group name you want to make this hack. This hack combined with specifying which **`TTY`** and **`VC`** devices super-user `root` is allowed to login on will improve your security a lot on the system.

Step 2
Now that we have defined the "`wheel`" group in our `/etc/pam.d/su` file configuration, it is time to add some users who will be allowed to `su` to super-user "`root`" account.

- If you want to make, for example, the user "`admin`" a member of the "`wheel`" group, and thus be able to `su` to `root`, use the following command:

 `[root@deep /]#` **`usermod -G10 admin`**

Which means "`G`" is a list of supplementary groups, where the user is also a member of. "`10`" is the numeric value of the user's `ID` "`wheel`", and "`admin`" is the user we want to add to the "`wheel`" group. Use the same command above for all users on your system you want to be able to `su` to super-user "`root`" account.

> **NOTE:** For Linux users, who use the Xwindow interface, it is important to note that if you can't `su` in a `GNOME` terminal, it's because you've used the wrong terminal. (So don't think that this advice doesn't work simply because of a `GNOME` terminal problem!)

Facultative:
A special line exists in the su file /etc/pam.d/su which allows you to implicitly trust users in the "wheel" group (for security reasons, I don't recommend using this option). This mean that all users who are members of the "wheel" group can su to root without the need to enter the super-user "root" password.

- To allow users who are members of the "wheel" group to su to root account without the need to enter the "root" password, edit the **su** file (vi /etc/pam.d/su) and uncomment the following line in the file:

```
auth      sufficient      /lib/security/pam_wheel.so trust use_uid
```

After this line has been uncommented, the /etc/pam.d/su file should look like this:

```
#%PAM-1.0
auth       sufficient   /lib/security/pam_rootok.so
auth       sufficient   /lib/security/pam_wheel.so trust use_uid
auth       required     /lib/security/pam_stack.so service=system-auth
account    required     /lib/security/pam_stack.so service=system-auth
password   required     /lib/security/pam_stack.so service=system-auth
session    required     /lib/security/pam_stack.so service=system-auth
session    optional     /lib/security/pam_xauth.so
```

Using sudo instead of su for logging as super-user

There is a security tool called "sudo" that we discuss into this book. This security software allow us to archive the same result as using the su command to get root privilege on the server but in a more secure and informative way. With sudo installed in our server, we can get information about who is connected as super-user root as well as many other useful features. Please see the chapter related to this security program into this book for more information about sudo.

If you want to use sudo to allow and control which is allowed to logging as super-user root on your server, then you no longer need to use the su command of Linux to archive this task and we can remove the SUID bit on this command to completely disable su and use sudo.

This let us remove one more SUID bit on our secure server and have a more complete and powerful security software to control access to super-user root. This is the method I highly recommend you to use instead of the su command of Linux.

Step 1
To archive this result, we have to remove the SUID bit of the su command and install the sudo security software as explained further down in this book. This also implies that we don't need to modify the above su configuration file on our system. To recap, all we need to do is to remove the SUID bit on the su command, and install sudo in our server.

- To remove the SUID bit on the su binary, use the following command:
 `[root@deep /]# chmod a-s /bin/su`

CHAPTER 5

General Optimization

IN THIS CHAPTER

1. Static vs. shared libraries
2. The Glibc 2 library of Linux
3. Why Linux programs are distributed as source
4. Some misunderstanding in the compiler flags options
5. The gcc specs file
6. Striping all binaries and libraries files
7. Tuning IDE Hard Disk Performance

Linux General Optimization

Abstract
At this stage of your configuration, you should now have a Linux server optimally configured and secured. Our server contains the most essential package and programs installed to be able to work properly and the most essential general system security configuration. Before we continue and begin to install the services we want to share with our customers, it is important to tune our Linux server to make it runs faster.

The tuning we will perform in the following part will be applied to the whole system. It also applies to present as well as future programs, such as services that we will later install. Generally, if you don't use an `x386` Intel processor, Red Hat Linux out of the box is not optimized for your specific CPU architecture (most people now run Linux on a Pentium processor). The sections below will guide you through different steps to optimize your Linux server for your specific processor, memory, and network.

Static vs. shared libraries
During compilation and build time of a program, the last stage (where all the parts of the program are joined together) is to link the software through the Linux libraries if needed. These libraries, which come in both shared and static formats, contain common system code which are kept in one place and shared between programs. Obviously there are some tasks that many programs will want to do, like opening files, and the codes that perform these functions are provided by the Linux libraries. On many Linux system these libraries files can be found into the `/lib`, `/usr/lib`, and `/usr/share` directories. The default behavior of Linux is to link shared and if it cannot find the shared libraries, then is to link statically.

One of the differences between using static or shared libraries are: When using a static library, the linker finds the bits that the program modules need, and directly copies them into the executable output file that it generates. For shared libraries, it leaves a note in the output saying, "When this program is run, it will first have to load this library".

Performance-wise, for most systems, worrying about static vs. dynamic is a moot point. There simply isn't enough difference to measure.

Security-wise there are valid arguments both ways. Static linking is less secure because it locks in the library bugs; unless you rebuild all such programs, your system won't be properly secured. Static linking is more secure because it avoids library attacks. The choice is yours: run a daemon which will remain vulnerable to library attacks, or run one which remains vulnerable to library bugs.

Portability-wise, the only difference is the size of the file you'll be transferring between systems.

To make setup easier, a statically linked daemon is only needed when the libraries are completely unavailable. That is rarely the case. Finally, on a busy system (when performance becomes a true issue), by statically linking you'll be DEGRADING performance. Being bigger, as more and more statically linked daemons are running, your system begins to swap sooner and since none of the code is shared, swapping will have a larger effect on performance. So, when looking to improve performance, you'll want to use shared libraries as much as possible.
<Gregory A Lundberg>

If you decide to compile program statically, you will generally need to add the "**-static**" and/or "**--disable-shared**" options flag to your compile line during compilation of your software. Be aware that it is not always possible to use and compile statically all programs, this highly depends on how developers are coding and developed the software.

To resume:

1. If you want to compile program with shared libraries, you will use something like:
```
CFLAGS="-O2 -march=i686 -funroll-loops"; export CFLAGS
./Configure \
```

2. If you want to compile program with static libraries, you will use something like:
```
CFLAGS="-O2 -static -march=i686 -funroll-loops"; export CFLAGS
./Configure \
--disable-shared \
```

WARNING: On Linux, static libraries have names like `libc.a`, while shared libraries are called `libc.so.x.y.z` where `x.y.z` is some form of version number since it would be quite a pain to recompile programs each time the version number changed so instead programs reference libraries by these shorter names and depend on the dynamic linker to make these shorter names symlinks to the current version. Shared libraries often have links pointing to them.

The `Glibc 2.2` library of Linux

The `Glibc 2.2`, which replaces the `libc4` and `libc5` that came before it, is the latest version of the `GNU C` Library for Linux and it contains standard libraries used by multiple programs on the system as described in the previous section. This particular package contains the most important sets of shared and static libraries, which provides the core functionality for `C` programs to run and without it, a Linux system would not function.

Under Red Hat Linux and most other Linux variant this package comes configured to run under i386 processor for portability reasons and this will pose problems for us if we want to compile programs under Linux because even if we have put in all the optimization flags we need to improve the speed of our server, when the compiler includes static or shared libraries files to our program, these library files will run optimized for an i386 processor.

In this case, our program will have some parts of its binaries optimized for an i686 processor (the program itself) and another parts optimized for an i386 processor (the `GLIBC` libraries). To solve the problem, you have to check inside your vendor CD-ROM media for available `GLIBC RPM` packages made to run on i686 `CPU` architecture. All vendors known about this issue and provide alternative `GLIBC` packages for i686 processors.

Why Linux programs are distributed as source

Linux has been ported to run on a large number of different machines and rather than provide a copy for each machine Linux can run on, it's much simpler just to distribute the source and let the end user compile it.

The creators of the distribution have no idea if you're going to be running it on a 386 or on a Pentium III and above so they have to write programs that work on all processors and this is where the problem comes, because all the programs that were installed with your distribution are going to be compiled so they work on the 386 for portability, meaning that they don't use any new feature like MMX which can only be found on newer generation of processors.

Fortunately, various compiler options exist to optimize program you want to install under Linux for your specific `CPU` architecture. This is great for those of us that want to tweak every ounce of performance out of the program, now we get to decide how the program is compiled. If you want some speed out of your programs you've got to know a fair amount about the various option flags you can use to compile.

The first thing you want to set is your `CPU` type, that's done with the "**`-march=cpu_type`**" (processor machine architecture) flag, an example would be "**`-march=i686`**" or "**`-march=k6`**", this will allow the compiler to select the appropriate optimizations for the processor, but this is only the beginning of what can be done.

You can set the "**`-o`**" flag anywhere from 1 to 3 to tell the compiler how aggressive to be with the optimizations, "**`-O3`**" will produce the fastest programs assuming the compiler didn't optimize an important part of a subroutine out. The next thing you might want to do is check out the "**`-f`**" options of the compiler, these are things like "**`-funroll-loops`**", and "**`-fomit-frame-pointer`**".

> **WARNING:** Compiling with the "**`-fomit-frame-pointer`**" switch option will use the stack for accessing variables. Unfortunately, debugging is almost impossible with this option. Also take special attention to the above optimization number "`-O3`"; "`O`" is a capital `o` and not a `0 (zero)`.

What I recommend you to use for all software that you want to compile on your server is the following optimizations FLAGS:

```
CFLAGS="-O2 -march=i686 -funroll-loops"
```

As you can see, I don't use the "`-O3`" and "`-fomit-frame-pointer`" options because some software have problem to run with these optimization options.

Some misunderstanding in the compiler flags options

At lot of discussions exist in the Linux community about the "**-O**" option and its level numbers. Some Linux users try to convince that level number up to "**-O3**" like "**-O9**" will produce faster program. The "**-O9**" flag doesn't do anything over "**-O3**", if you don't believe me make a small file, call it `testO3.c` and see:

Step 1

- Create the `testO3.c` file with the following command:
 `[root@deep tmp]#` **`touch testO3.c`**

Step 2

- Run the `GCC` compiler with "**-O3**" flag through the `testO3.c` file with the command:
 `[root@deep tmp]#` **`gcc -O3 -S -fverbose-asm testO3.c`**

Step 3

Look at `testO3.s` that it made, then run again with "**-O9**" and compare the output.

- Create the `testO9.c` file with the following command:
 `[root@deep tmp]#` **`touch testO9.c`**

Step 4

- Run the `GCC` compiler again with "**-O9**" flag through the `testO9.c` file with command:
 `[root@deep tmp]#` **`gcc -O9 -S -fverbose-asm testO9.c`**

Step 5

Now if you compare the output you will see no difference between the both files.

- To compare the output, use the following command:
 `[root@deep tmp]#` **`diff testO3.s testO9.s > difference`**

> **WARNING:** The "**-O3**" flag level number is the best and highest optimization flag you can use during optimization of programs under Linux.

The gcc specs file

The /usr/lib/gcc-lib/i386-redhat-linux/2.96/specs file of Red Hat Linux is a set of defines that the gcc compiler uses internally to set various aspects of the compile environment. All customizations that you put in this file will apply for the entire variable environment on your system, so putting optimization flags in this file is a good choice.

To squeeze the maximum performance from your x86 programs, you can use full optimization when compiling with the "**-O3**" flag. Many programs contain "**-O2**" in the Makefile. The "**-O3**" level number is the highest level of optimization. It will increase the size of what it produces, but it runs faster in most case. You can also use the "**-march=cpu_type**" switch to optimize the program for the CPU listed to the best of GCC's ability. However, the resulting code will only be run able on the indicated CPU or higher.

Below are the optimization flags that **we recommend** you to put in your /usr/lib/gcc-lib/i386-redhat-linux/2.96/specs file depending on your CPU architecture. The optimization options apply only when we compile and install a new program in our server. These optimizations don't play any role in our Linux base system; it just tells our compiler to optimize the new programs that we will install with the optimization flags we have specified in the /usr/lib/gcc-lib/i386-redhat-linux/2.96/specs file. Adding options listed below depending of your CPU architecture to the gcc 2.96 specs file will save you having to change every CFLAGS in future Makefiles.

Step 1
The first thing to do is to verify the compiler version installed on your Linux server.

- To verify the compiler version installed on your system, use the command:

```
[root@deep /]#  gcc -v
Reading specs from /usr/lib/gcc-lib/i386-redhat-linux/2.96/specs
gcc version 2.96 20000731 (Red Hat Linux 7.3 2.96-110)
```

Step 2

For CPU i686 or PentiumPro, Pentium II, Pentium III, and Athlon

Edit the /usr/lib/gcc-lib/i386-redhat-linux/2.96/specs file, scroll down a ways... You'll see a section like the following:

```
*cpp_cpu_default:
-D__tune_i386__

*cpp_cpu:
-Acpu(i386) -Amachine(i386) %{!ansi:-Di386} -D__i386 -D__i386__
%{march=i386:%{!mcpu*:-D__tune_i386__ }}%{march=i486:-D__i486 -D__i486__
%{!mcpu*:-D__tune_i486__ }}%{march=pentium|march=i586:-D__pentium -D__pentium__
%{!mcpu*:-D__tune_pentium__ }}%{march=pentiumpro|march=i686:-D__pentiumpro -
D__pentiumpro__    %{!mcpu*:-D__tune_pentiumpro__ }}%{march=k6:-D__k6 -D__k6__
%{!mcpu*:-D__tune_k6__ }}%{march=athlon:-D__athlon -D__athlon__ %{!mcpu*:-
D__tune_athlon__ }}%{m386|mcpu=i386:-D__tune_i386__ }%{m486|mcpu=i486:-
D__tune_i486__ }%{mpentium|mcpu=pentium|mcpu=i586:-D__tune_pentium__
}%{mpentiumpro|mcpu=pentiumpro|mcpu=i686:-D__tune_pentiumpro__ }%{mcpu=k6:-
D__tune_k6__ }%{mcpu=athlon:-D__tune_athlon__
}%{!march*:%{!mcpu*:%{!m386:%{!m486:%{!mpentium*:%(cpp_cpu_default)}}}}}

*cc1_cpu:
%{!mcpu*: %{m386:-mcpu=i386} %{m486:-mcpu=i486} %{mpentium:-mcpu=pentium}
%{mpentiumpro:-mcpu=pentiumpro}}
```

Change it for the following:

```
*cpp_cpu_default:
-D__tune_i686__

*cpp_cpu:
-Acpu(i386) -Amachine(i386) %{!ansi:-Di386} -D__i386 -D__i386__
%{march=i386:%{!mcpu*:-D__tune_i386__ }}%{march=i486:-D__i486 -D__i486__
%{!mcpu*:-D__tune_i486__ }}%{march=pentium|march=i586:-D__pentium -D__pentium__
%{!mcpu*:-D__tune_pentium__ }}%{march=pentiumpro|march=i686:-D__pentiumpro -
D__pentiumpro__   %{!mcpu*:-D__tune_pentiumpro__ }}%{march=k6:-D__k6 -D__k6__
%{!mcpu*:-D__tune_k6__ }}%{march=athlon:-D__athlon -D__athlon__ %{!mcpu*:-
D__tune_athlon__ }}%{m386|mcpu=i386:-D__tune_i386__ }%{m486|mcpu=i486:-
D__tune_i486__ }%{mpentium|mcpu=pentium|mcpu=i586:-D__tune_pentium__
}%{mpentiumpro|mcpu=pentiumpro|mcpu=i686:-D__tune_pentiumpro__ }%{mcpu=k6:-
D__tune_k6__ }%{mcpu=athlon:-D__tune_athlon__
}%{!march*:%{!mcpu*:%{!m386:%{!m486:%{!mpentium*:%(cpp_cpu_default)}}}}}

*cc1_cpu:
%{!mcpu*: -O2 -march=i686 -funroll-loops %{m386:-mcpu=i386} %{m486:-mcpu=i486}
%{mpentium:-mcpu=pentium} %{mpentiumpro:-mcpu=pentiumpro}}
```

WARNING: Make sure that you're putting `-O2` and not `-02` (dash zero three).

For CPU i586 or Pentium

Edit the `/usr/lib/gcc-lib/i386-redhat-linux/2.96/specs` file, scroll down a ways... You'll see a section like the following:

```
*cpp_cpu_default:
-D__tune_i386__

*cpp_cpu:
-Acpu(i386) -Amachine(i386) %{!ansi:-Di386} -D__i386 -D__i386__
%{march=i386:%{!mcpu*:-D__tune_i386__ }}%{march=i486:-D__i486 -D__i486__
%{!mcpu*:-D__tune_i486__ }}%{march=pentium|march=i586:-D__pentium -D__pentium__
%{!mcpu*:-D__tune_pentium__ }}%{march=pentiumpro|march=i686:-D__pentiumpro -
D__pentiumpro__   %{!mcpu*:-D__tune_pentiumpro__ }}%{march=k6:-D__k6 -D__k6__
%{!mcpu*:-D__tune_k6__ }}%{march=athlon:-D__athlon -D__athlon__ %{!mcpu*:-
D__tune_athlon__ }}%{m386|mcpu=i386:-D__tune_i386__ }%{m486|mcpu=i486:-
D__tune_i486__ }%{mpentium|mcpu=pentium|mcpu=i586:-D__tune_pentium__
}%{mpentiumpro|mcpu=pentiumpro|mcpu=i686:-D__tune_pentiumpro__ }%{mcpu=k6:-
D__tune_k6__ }%{mcpu=athlon:-D__tune_athlon__
}%{!march*:%{!mcpu*:%{!m386:%{!m486:%{!mpentium*:%(cpp_cpu_default)}}}}}

*cc1_cpu:
%{!mcpu*: %{m386:-mcpu=i386} %{m486:-mcpu=i486} %{mpentium:-mcpu=pentium}
%{mpentiumpro:-mcpu=pentiumpro}}
```

Change it for the following:

```
*cpp_cpu_default:
-D__tune_i586__

*cpp_cpu:
-Acpu(i386) -Amachine(i386) %{!ansi:-Di386} -D__i386 -D__i386__
%{march=i386:%{!mcpu*:-D__tune_i386__ }}%{march=i486:-D__i486 -D__i486__
%{!mcpu*:-D__tune_i486__ }}%{march=pentium|march=i586:-D__pentium -D__pentium__
%{!mcpu*:-D__tune_pentium__ }}%{march=pentiumpro|march=i686:-D__pentiumpro -
```

```
D__pentiumpro__   %{!mcpu*:-D__tune_pentiumpro__ }}%{march=k6:-D__k6 -D__k6__
%{!mcpu*:-D__tune_k6__ }}%{march=athlon:-D__athlon -D__athlon__ %{!mcpu*:-
D__tune_athlon__ }}%{m386|mcpu=i386:-D__tune_i386__ }%{m486|mcpu=i486:-
D__tune_i486__ }%{mpentium|mcpu=pentium|mcpu=i586:-D__tune_pentium__
}%{mpentiumpro|mcpu=pentiumpro|mcpu=i686:-D__tune_pentiumpro__ }%{mcpu=k6:-
D__tune_k6__ }%{mcpu=athlon:-D__tune_athlon__
}%{!march*:%{!mcpu*:%{!m386:%{!m486:%{!mpentium*:%(cpp_cpu_default)}}}}}

*cc1_cpu:
%{!mcpu*: -O2 -march=i586 -funroll-loops %{m386:-mcpu=i386} %{m486:-mcpu=i486}
%{mpentium:-mcpu=pentium} %{mpentiumpro:-mcpu=pentiumpro}}
```

WARNING: Make sure that you're putting `-O2` and not `-02` (dash zero three).

For CPU i486

Edit the `/usr/lib/gcc-lib/i386-redhat-linux/2.96/specs` file, scroll down a ways... You'll see a section like the following:

```
*cpp_cpu_default:
-D__tune_i386__

*cpp_cpu:
-Acpu(i386) -Amachine(i386) %{!ansi:-Di386} -D__i386 -D__i386__
%{march=i386:%{!mcpu*:-D__tune_i386__ }}%{march=i486:-D__i486 -D__i486__
%{!mcpu*:-D__tune_i486__ }}%{march=pentium|march=i586:-D__pentium -D__pentium__
%{!mcpu*:-D__tune_pentium__ }}%{march=pentiumpro|march=i686:-D__pentiumpro -
D__pentiumpro__   %{!mcpu*:-D__tune_pentiumpro__ }}%{march=k6:-D__k6 -D__k6__
%{!mcpu*:-D__tune_k6__ }}%{march=athlon:-D__athlon -D__athlon__ %{!mcpu*:-
D__tune_athlon__ }}%{m386|mcpu=i386:-D__tune_i386__ }%{m486|mcpu=i486:-
D__tune_i486__ }%{mpentium|mcpu=pentium|mcpu=i586:-D__tune_pentium__
}%{mpentiumpro|mcpu=pentiumpro|mcpu=i686:-D__tune_pentiumpro__ }%{mcpu=k6:-
D__tune_k6__ }%{mcpu=athlon:-D__tune_athlon__
}%{!march*:%{!mcpu*:%{!m386:%{!m486:%{!mpentium*:%(cpp_cpu_default)}}}}}

*cc1_cpu:
%{!mcpu*: %{m386:-mcpu=i386} %{m486:-mcpu=i486} %{mpentium:-mcpu=pentium}
%{mpentiumpro:-mcpu=pentiumpro}}
```

Change it for the following:

```
*cpp_cpu_default:
-D__tune_i486__

*cpp_cpu:
-Acpu(i386) -Amachine(i386) %{!ansi:-Di386} -D__i386 -D__i386__
%{march=i386:%{!mcpu*:-D__tune_i386__ }}%{march=i486:-D__i486 -D__i486__
%{!mcpu*:-D__tune_i486__ }}%{march=pentium|march=i586:-D__pentium -D__pentium__
%{!mcpu*:-D__tune_pentium__ }}%{march=pentiumpro|march=i686:-D__pentiumpro -
D__pentiumpro__   %{!mcpu*:-D__tune_pentiumpro__ }}%{march=k6:-D__k6 -D__k6__
%{!mcpu*:-D__tune_k6__ }}%{march=athlon:-D__athlon -D__athlon__ %{!mcpu*:-
D__tune_athlon__ }}%{m386|mcpu=i386:-D__tune_i386__ }%{m486|mcpu=i486:-
D__tune_i486__ }%{mpentium|mcpu=pentium|mcpu=i586:-D__tune_pentium__
}%{mpentiumpro|mcpu=pentiumpro|mcpu=i686:-D__tune_pentiumpro__ }%{mcpu=k6:-
D__tune_k6__ }%{mcpu=athlon:-D__tune_athlon__
}%{!march*:%{!mcpu*:%{!m386:%{!m486:%{!mpentium*:%(cpp_cpu_default)}}}}}

*cc1_cpu:
```

```
%{!mcpu*: -O2 -march=i486 -funroll-loops %{m386:-mcpu=i386} %{m486:-mcpu=i486}
%{mpentium:-mcpu=pentium} %{mpentiumpro:-mcpu=pentiumpro}}
```

WARNING: Make sure that you're putting `-O2` and not `-02` (dash zero three).

For CPU AMD K6 or K6-2

Edit the `/usr/lib/gcc-lib/i386-redhat-linux/2.96/specs` file, scroll down a ways... You'll see a section like the following:

```
*cpp_cpu_default:
-D__tune_i386__

*cpp_cpu:
-Acpu(i386) -Amachine(i386) %{!ansi:-Di386} -D__i386 -D__i386__
%{march=i386:%{!mcpu*:-D__tune_i386__ }}%{march=i486:-D__i486 -D__i486__
%{!mcpu*:-D__tune_i486__ }}%{march=pentium|march=i586:-D__pentium -D__pentium__
%{!mcpu*:-D__tune_pentium__ }}%{march=pentiumpro|march=i686:-D__pentiumpro -
D__pentiumpro__    %{!mcpu*:-D__tune_pentiumpro__ }}%{march=k6:-D__k6 -D__k6__
%{!mcpu*:-D__tune_k6__ }}%{march=athlon:-D__athlon -D__athlon__ %{!mcpu*:-
D__tune_athlon__ }}%{m386|mcpu=i386:-D__tune_i386__ }%{m486|mcpu=i486:-
D__tune_i486__ }%{mpentium|mcpu=pentium|mcpu=i586:-D__tune_pentium__
}%{mpentiumpro|mcpu=pentiumpro|mcpu=i686:-D__tune_pentiumpro__ }%{mcpu=k6:-
D__tune_k6__ }%{mcpu=athlon:-D__tune_athlon__
}%{!march*:%{!mcpu*:%{!m386:%{!m486:%{!mpentium*:%(cpp_cpu_default)}}}}}

*cc1_cpu:
%{!mcpu*: %{m386:-mcpu=i386} %{m486:-mcpu=i486} %{mpentium:-mcpu=pentium}
%{mpentiumpro:-mcpu=pentiumpro}}
```

Change it for the following:

```
*cpp_cpu_default:
-D__tune_k6__

*cpp_cpu:
-Acpu(i386) -Amachine(i386) %{!ansi:-Di386} -D__i386 -D__i386__
%{march=i386:%{!mcpu*:-D__tune_i386__ }}%{march=i486:-D__i486 -D__i486__
%{!mcpu*:-D__tune_i486__ }}%{march=pentium|march=i586:-D__pentium -D__pentium__
%{!mcpu*:-D__tune_pentium__ }}%{march=pentiumpro|march=i686:-D__pentiumpro -
D__pentiumpro__    %{!mcpu*:-D__tune_pentiumpro__ }}%{march=k6:-D__k6 -D__k6__
%{!mcpu*:-D__tune_k6__ }}%{march=athlon:-D__athlon -D__athlon__ %{!mcpu*:-
D__tune_athlon__ }}%{m386|mcpu=i386:-D__tune_i386__ }%{m486|mcpu=i486:-
D__tune_i486__ }%{mpentium|mcpu=pentium|mcpu=i586:-D__tune_pentium__
}%{mpentiumpro|mcpu=pentiumpro|mcpu=i686:-D__tune_pentiumpro__ }%{mcpu=k6:-
D__tune_k6__ }%{mcpu=athlon:-D__tune_athlon__
}%{!march*:%{!mcpu*:%{!m386:%{!m486:%{!mpentium*:%(cpp_cpu_default)}}}}}

*cc1_cpu:
%{!mcpu*: -O2 -march=k6 -funroll-loops %{m386:-mcpu=i386} %{m486:-mcpu=i486}
%{mpentium:-mcpu=pentium} %{mpentiumpro:-mcpu=pentiumpro}}
```

WARNING: Make sure that you're putting `-O2` and not `-02` (dash zero three).

Step3
Once our optimization flags have been applied to the `gcc 2.96` specs file, it time to verify if the modification work.

- To verify if the optimization work, use the following commands:
```
[root@deep tmp]# touch cpu.c
[root@deep tmp]# gcc cpu.c -S -fverbose-asm
[root@deep tmp]# less cpu.s
```

What you'll get is a file that contains depending of options you have chose, something like:

```
        .file   "ccnVPjeW.i"
        .version        "01.01"
# GNU C version 2.96 20000731 (Red Hat Linux 7.3 2.96-110) (i386-redhat-linux)
compiled by GNU C version 2.96 20000731 (Red Hat Linux 7.3 2.96-110).
# options passed:  -O2 -march=i686 -funroll-loops -fverbose-asm
# options enabled:  -fdefer-pop -foptimize-sibling-calls -fcse-follow-jumps
# -fcse-skip-blocks -fexpensive-optimizations -fthread-jumps
# -fstrength-reduce -funroll-loops -fpeephole -fforce-mem -ffunction-cse
# -finline -fkeep-static-consts -fcaller-saves -fpcc-struct-return -fgcse
# -frerun-cse-after-loop -frerun-loop-opt -fdelete-null-pointer-checks
# -fschedule-insns2 -fsched-interblock -fsched-spec -fbranch-count-reg
# -fnew-exceptions -fcommon -fverbose-asm -fgnu-linker -fregmove
# -foptimize-register-move -fargument-alias -fstrict-aliasing
# -fmerge-constants -fident -fpeephole2 -fmath-errno -m80387 -mhard-float
# -mno-soft-float -mieee-fp -mfp-ret-in-387 -march=i686

gcc2_compiled.:
        .ident  "GCC: (GNU) 2.96 20000731 (Red Hat Linux 7.3 2.96-110)"
```

WARNING: In our example we are optimized the specs file for a i686 `CPU` processor. It is important to note that most of the "**`-f`**" options are automatically included when you use "**-O2**" and don't need to be specified again. The changes that were shown were made so that a command like "**`gcc`**" would really be the command "**`gcc -march=i686`**" without having to change every single `Makefile` which can really be a pain.

Below is the explanation of the different optimization options we use:

- **The "`-march=cpu_type`" optimization flag**
 The "`-march=cpu_type`" optimization option will set the default `CPU` to use for the machine type when scheduling instructions.

- **The "`-funroll-loops`" optimization flag**
 The "`-funroll-loops`" optimization option will perform the optimization of loop unrolling and will do it only for loops whose number of iterations can be determined at compile time or run time.

- **The "`-fomit-frame-pointer`" optimization flag**
 The "`-fomit-frame-pointer`" optimization option, one of the most interesting, will allow the program to not keep the frame pointer in a register for functions that don't need one. This avoids the instructions to save, set up and restores frame pointers; it also makes an extra register available in many functions and makes debugging impossible on most machines.

WARNING: All future optimizations that we will describe in this book refer by default to a Pentium PRO/II/III and higher i686 `CPU` family. So you must adjust the compilation flags for your specific `CPU` processor type in the `/usr/lib/gcc-lib/i386-redhat-linux/2.96/specs` file and during your compilation time.

Striping all binaries and libraries files

When compiler builds program it add many comments and other stuffs inside the resulting binary and library code which are used to debug application on the system or to make the code more readable by developers. To get the latest bit of optimization, we can remove all comments and other unneeded stuffs since there are only used for debugging purpose. This will make the software runs a little bit faster because it will have fewer codes to read when executing.

I don't know if it's a good idea to talk about this hack because it's really dangerous to apply and can make your system unstable or completely unworkable if you don't take care of what you do. The process of eliminating all unneeded comments and other unneeded stuffs from your binaries and libraries files is made by the use of the `strip` command of Linux. This command should be used with care and in the good manner or you will certainly have a bad surprise.

Bellow, I will explain you how to apply it on your system and on which files or where you should use it. It is very important to know that it's NOT all binaries and especially libraries files that need to be striped by this method but ONLY some of them. If you apply this hack on your entire system, then something will inevitably break, you have been warned.

Finally, you should use this hack on servers where you DON'T compile software. If you compile software on the server where you want to apply this hack, then nothing will work and you will not be able to compile any software on it. Use this hack on server which doesn't have any compiler packages installed to make compilation.

Step 1
The first step in our procedure will be to be sure that the `strip` command is available on our server; this command comes from the "`binutils`" `RPM` package. Therefore, if it is not installed, install it from your CD-ROM.

Step 2
Once the `strip` program is installed on our server, it's time to strip the required files. With the commands below, we strip all binaries program available under the `/bin`, `/sbin`, `/usr/bin` and `/usr/sbin` directories of your server.

- To strip all binaries program, use the following commands:

```
[root@deep /]# strip /bin/*
[root@deep /]# strip /sbin/*
[root@deep /]# strip /usr/bin/*
[root@deep /]# strip /usr/sbin/*
```

NOTE: When issuing the above commands, you will receive some error messages like "`File format not recognized`" on your terminal. This is normal because some of the binaries files are symbolic link pointing to other binaries on your system and the `strip` command generate the warning because it cannot strip symbolic links.

Step 3
Now, it's time to strip the libraries files. This is where the action can become dangerous if you don't take care or abuse the `strip` command. With the commands below, we strip all libraries files available under the `/lib` and `/usr/lib` directory of the system.

- To strip all libraries files, use the following command:
```
[root@deep /]# strip -R .comment /usr/lib/*.so.*
[root@deep /]# strip -R .comment /lib/*.so.*
```

NOTE: Make attention to the above command, you can see here that I use the "`-R`" option with the `strip` command. This option allows us to select a specific name to strip from the target libraries files. With the "`.comment`" name, we inform the command to remove any lines inside the libraries codes where this name appears. You can see that I don't use the `strip` command without any option as I do for the above step related to binaries program. This is very important and you should never use the `strip` command without the above option to strip libraries files on your system.

Tuning `IDE` Hard Disk Performance

Accessing a hard disk can be 50 to 100 times slower than reading data from `RAM`. File caches using `RAM` can alleviate this. However, low memory conditions will reduce the amount of memory available for the file-system cache slowing things down. File systems can also become heavily fragmented, slowing down disk accesses. Heavy use of symbolic links on Unix systems can slow down disk accesses too.

Default Linux installs are also notorious for setting hard disk default settings which are tuned for compatibility and not for speed. Use the command `hdparm` to tune your Linux hard disk settings.

The `hdparm` is a tool, which can be used to tune and improve the performance of your `IDE` hard disk. By default, any `IDE` drives you have in your Linux system are not optimized. Even if you have an `ULTRA DMA` system you will not be able to take full advantage of its speed if you are not using the `hdparm` tool to enable its features. This is because there is many different hard drive makes and models and Linux cannot know every feature of each one.

Performance increases have been reported on massive disk `I/O` operations by setting the `IDE` drivers to use `DMA`, 32-bit transfers and multiple sector modes. The kernel seems to use more conservative settings unless told otherwise. The magic command to change the setting of your drive is `hdparm`.

Before going into the optimization of your hard drive, it is important to verify that the `hdparm` package is installed in your system. If you have followed every step during the installation of Linux on your computer, then this package is not installed.

- To verify if `hdparm` package is installed on your system, use the command:
```
[root@deep /]# rpm -q hdparm
package hdparm is not installed
```

If the `hdparm` package seems not to be installed, you'll need to mount your CD-ROM drive containing the Linux CD-ROM Part 1 and install it.

- To mount the CD-ROM drive, use the following commands:
```
[root@deep /]# mount /dev/cdrom /mnt/cdrom/
had: ATAPI 32X CD-ROM drive, 128kB Cache
mount: block device dev/cdrom is write-protected, mounting read-only
```

- To install the hdparm package on your Linux system, use the following command:
```
[root@deep /]# cd /mnt/cdrom/RedHat/RPMS/
[root@deep RPMS]# rpm -Uvh hdparm-version.i386.rpm
hdparm       ##################################################
```

- To unmount your CD-ROM drive, use the following command:
```
[root@deep RPMS]# cd /; umount /mnt/cdrom/
```

Once hdparm package is installed on the system, it is time to go into the optimization of your hard drive. It is important to note that depending on your model and make, there will be some parameters that will apply and other that don't. It is to your responsibility to know and understand your disk drive before applying any optimization parameters as described below.

Finally, and especially for UltraDMA systems, it is vital to verify under your BIOS settings if the parameters related to DMA support on your computer are enabled or you will inevitably break your hard disk. You have been warned.

Step 1
The first parameter applies to the majority of all modern drives and models in the market and enables 32-bit I/O over PCI buses. This option is one of the most important and will usually double the speed of your drive.

- To enable 32-bit I/O over the PCI buses, use the following command:
```
[root@deep /]#  /sbin/hdparm -c3 /dev/hda (or hdb, hdc etc).
```

This will usually, depending on your IDE Disk Drive model, cut the timing buffered disk reads time by two. The hdparm (8) manpage says that you may need to use "**-c3**" for many chipsets since it works with nearly all 32-bit IDE chipsets. All (E)IDE drives still have only a 16-bit connection over the ribbon cable from the interface card.

Step 2
The second parameter applies only on standard DMA disk and will activate the simple DMA feature of the disk. This feature is for old disk drives with DMA capabilities.

- To enable DMA, use the following command:
```
[root@deep /]#  /sbin/hdparm -d1 /dev/hda (or hdb, hdc etc).
```

This may depend on support for your motherboard chipset being compiled into your kernel. Also, this command will enable DMA support for your hard drive only for interfaces which support DMA, it will cut the timing buffered disk reads time and will improve the performance by two.

Step 3
Multiword DMA mode 2, also known as ATA2 disk drive is the successor of the simple DMA drive. If you have this kind of hard drive, then you must enable the parameter in your Linux system.

- To enable multiword DMA mode 2 transfers, use the following command:
 `[root@deep /]#  /sbin/hdparm -d1 -X34 /dev/hda (or hdb, hdc etc).`

This sets the IDE transfer mode for newer (E)IDE/ATA2 drives. (Check your hardware manual to see if you have it).

Step 4
As for DMA mode 2, the UltraDMA mode 2 is an improvement of the DMA technology. If you have this kind of drive in your system, then choose this mode.

- To enable UltraDMA mode 2 transfers, use the following command:
 `[root@deep /]#  /sbin/hdparm -d1 -X66 /dev/hda (or hdb, hdc etc)`

See your manual page about hdparm for more information. USE THIS OPTION WITH EXTREME CAUTION!

Step 5
The UltraDMA mode 4 is one of the latest entries and one of the most popular at this time; it is also known and referred as ATA/66. I guess that most of you have this kind of drive installed and if it is the case then it is the one that you must choose for sure.

- To enable UltraDMA mode4 transfers, use the following command:
 `[root@deep /]#  /sbin/hdparm -d1 -X12 -X68 /dev/hda (or hdb, hdc etc)`

This will enable UltraDMA ATA/66 mode on your drive. See your manual page about hdparm for more information. USE THIS OPTION WITH EXTREME CAUTION!

Step 6
Multiple sector mode (aka IDE Block Mode), is a feature of most modern IDE hard drives, permitting the transfer of multiple sectors per I/O interrupt, rather than the usual one sector per interrupt. When this feature is enabled, it typically reduces operating system overhead for disk I/O by 30-50%. On many systems it also provides increased data throughput of anywhere from 5% to 50%.

- To set multiple sector mode I/O, use the following command:
 `[root@deep /]#  /sbin/hdparm -mXX /dev/hda (or hdb, hdc etc)`

Where "XX" represent the maximum setting supported by your drive. The "**-i**" flag can be used to find the maximum setting supported by an installed drive: look for **MaxMultSect** in the output.

- To find the maximum setting of your drive, use the following command:
 `[root@deep /]#` **`/sbin/hdparm -i /dev/hda`** `(or hdb, hdc etc)`

```
/dev/hda:

 Model=QUANTUM FIREBALLP LM15, FwRev=A35.0700, SerialNo=883012661990
 Config={ HardSect NotMFM HdSw>15uSec Fixed DTR>10Mbs }
 RawCHS=16383/16/63, TrkSize=32256, SectSize=21298, ECCbytes=4
 BuffType=3(DualPortCache), BuffSize=1900kB, MaxMultSect=16, MultSect=16
 DblWordIO=no, OldPIO=2, DMA=yes, OldDMA=2
 CurCHS=16383/16/63, CurSects=-66060037, LBA=yes, LBAsects=29336832
 tDMA={min:120,rec:120}, DMA modes: mword0 mword1 mword2
 IORDY=on/off, tPIO={min:120,w/IORDY:120}, PIO modes: mode3 mode4
 UDMA modes: mode0 mode1 mode2 mode3 *mode4
```

Step 7
The get/set sector count is used to improve performance in sequential reads of large files! The default setting is 8 sectors (4KB) and we will double and change it for 16. USE THIS OPTION WITH EXTREME CAUTION!

- To improve the get/set sector count for file system read-ahead, use the command:
 `[root@deep /]#` **`/sbin/hdparm -a16 /dev/hda`** `(or hdb, hdc etc)`

Step 8
The get/set interrupt-unmask flag will greatly improve Linux's responsiveness and eliminates "serial port overrun" errors. USE THIS OPTION WITH EXTREME CAUTION!

- To improve and get/set interrupt-unmask flag for the drive, use the command:
 `[root@deep /]#` **`/sbin/hdparm -u1 /dev/hda`** `(or hdb, hdc etc)`

Step 9
The `IDE` drive's write-caching feature will improve the performance of the hard disk. USE THIS OPTION WITH EXTREME CAUTION!

- To enable the `IDE` drive's write-caching feature, use the following command:
 `[root@deep /]#` **`/sbin/hdparm -W1 /dev/hda`** `(or hdb, hdc etc)`

Step 10
These options will allow the drive to retain your settings over a soft reset (as done during the error recovery sequence). It is important to note that not all drives support this feature.

- To enables the drive to retain your settings, use the command:
 `[root@deep /]#` **`/sbin/hdparm -K1 -k1 /dev/hda`** `(or hdb, hdc etc)`

Step 11
Once every tuning related to your specific drive have been set, you can test the results and see if you want to keep them or not.

- You can test the results of your changes by running `hdparm` in performance test mode:
 `[root@deep /]#` **`/sbin/hdparm -vtT /dev/hda`** `(or hdb, hdc etc).`

```
/dev/hda:
 multcount       = 16 (on)
 I/O support     =  3 (32-bit w/sync)
 unmaskirq       =  1 (on)
 using_dma       =  1 (on)
 keepsettings    =  1 (on)
 nowerr          =  0 (off)
 readonly        =  0 (off)
 readahead       = 16 (on)
 geometry        = 1826/255/63, sectors = 29336832, start = 0
 Timing buffer-cache reads:  128 MB in  0.85 seconds = 150.59 MB/sec
 Timing buffered disk reads:  64 MB in  2.54 seconds = 25.20 MB/sec
```

Once you have a set of `hdparm` options, you can put the commands in your `/etc/rc.local` file to run it every time you reboot the machine. When running from `/etc/rc.local`, you can add the "**-q**" option for reducing screen clutter. In my case, I will put the following configuration in the end of my `rc.local` file:

```
/sbin/hdparm -q -c3 -d1 -X12 -X68 -m16 -a16 -u1 -W1 -k1 -K1 /dev/had
```

CHAPTER 6

Kernel Security & Optimization

IN THIS CHAPTER

1. **Difference between a Modularized Kernel and a Monolithic Kernel**
2. **Making an emergency boot floppy**
3. **Preparing the Kernel for the installation**
4. **Applying the `Grsecurity` kernel patch**
5. **Obtaining and Installing `Grsecurity`**
6. **Tuning the Kernel**
7. **Cleaning up the Kernel**
8. **Configuring the Kernel**
9. **Compiling the Kernel**
10. **Installing the Kernel**
11. **Verifying or upgrading your boot loader**
12. **Reconfiguring `/etc/modules.conf` file**
13. **Rebooting your system to load the new kernel**
14. **Delete programs, edit files pertaining to modules**
15. **Making a new rescue floppy for Modularized Kernel**
16. **Making a emergency boot floppy disk for Monolithic Kernel**

Linux Kernel

Abstract
Well, our Linux server seems to be getting in shape now! But wait, what is the most important part of our server? Yes, it's the kernel. The Linux kernel is the core of our operating system, and without it there is no Linux at all. So we must configure the kernel to fit our needs and compile only the features we really need.

The new generation of Linux Kernel 2.4 was seemingly written with the server in mind. Many of the old limits, which prevented Linux being adopted in the "enterprise" market, have been lifted. The first thing to do next is to build a kernel that best suits your system. It's very simple to do but, in any case, refer to the `README` file in the `/usr/src/linux` source directory after uncompressing the archive on your system. When configuring your kernel, only compile in code that you need. A few reasons that come to mind are:

- ✓ The Kernel will be faster (less code to run);
- ✓ You will have more memory (Kernel parts are NEVER swapped to the virtual memory);
- ✓ More stable (Ever probed for a non-existent card?);
- ✓ Unnecessary parts can be used by an attacker to gain access to the machine or other machines on the network.
- ✓ Modules are also slower than support compiled directly in the kernel.

In our configuration and compilation we will firstly show you how to build a `monolithic kernel`, which is the recommended method for better performance and security and a `modularized kernel` for easily portability between different Linux systems. `Monolithic kernel` means to only answer **yes** or **no** to the questions (don't make anything modular) and omits the steps: **`make modules`** and **`make modules_install`**.

Difference between a Modularized Kernel and a Monolithic Kernel
I don't want to go into deeply technical descriptions here. I'll try to stay as simple as I can in my explanation, this will allow us to better understand the differences. Firstly, it is evident that not all computers are identical; someone may have a new computer with the latest processor, a lot of memory, running on a SCSI sub-system with a good motherboard, where others may have an old computer with an older Pentium II processor, 128 MB of memory, on an IDE sub-system with a standard motherboard. These differences push kernel developers to constantly add or update for new drivers and features into the kernel code and are one of the reasons why a Modularized Kernel exists.

Without all of those differences, it would be simple to provide a kernel where all the drivers and features are already included, but this is impossible because we all have different computers. Someone may say: "ok we can include all presently available drivers and features into the kernel and it will run on any computer". This approach poses some problems. Firstly, it will make the kernel binary bigger and slower. Secondly, the Kernel will probe for nonexistent hardware, features and maintenance of other programs that directly depend on the kernel and will become more complicated.

A solution was found and this was the Modularized Kernel approach. A technique that allows small pieces of compiled code to be inserted in or removed from the running kernel. In this way the Kernel will only load and run drivers and features that your computer have and will forget about the others. This practice is what all Linux vendors use to provide Linux kernels. They build and link every driver and feature as a module (which keeps the binary kernel smaller) that can be recognized and loaded if, and only if, they are needed by the kernel or the system.

Kernel developers provide the ability to build a Modularized kernel, through an option that asks you during kernel configuration if you want to build the available drivers/features as a module. This option appears at the beginning of the Kernel configuration in the following form "`Enable loadable module support (CONFIG_MODULES) [Y/n/?]`". If you answer "`Yes`" here, then the compiled Kernel will be a Modularized Kernel and all future questions appearing during kernel configuration will give you the choice to compile the drivers/features into the Kernel code as a module by answering "`m`" for module, "`y`" for yes includes the code, or "`n`" do not include the code. Alternatively, if you answer "`No`" to the question "`Enable loadable module support (CONFIG_MODULES) [Y/n/?]`", then the corresponding Kernel will be a Monolithic kernel and all future questions appearing during kernel configuration will let you answer either "`y`" (yes, include the driver/feature) or "`n`" (no, do not include the drivers/feature). This allows you to build a Kernel where every driver/feature is compiled into it.

To recap, Modularized Kernels allow small pieces of compiled code, which reside under the `/lib/modules/2.4.x-x/` kernel directory to be inserted into or removed from the running kernel and a Monolithic Kernel contains the drivers/features into its compiled code.

Some people will say that a loadable module is as good as hard-linked code. But what sort of speed difference is seen when using loadable modules instead of hard-linked code? Well, here's an extract of the kernel mailing list archive:

The immediate response from some was "almost nothing," but further consideration has shown this not to be true. There are, in fact, a number of costs associated with loadable modules. The biggest, perhaps, relates to how loadable modules are placed in kernel memory. The code for a module needs to live in a contiguous address space. The kernel sets up that address space with a function called `vmalloc`, which allocates memory with virtual addresses. In other words, a loadable module is in an address space that is visible to the kernel, but which is separate from where the core kernel code goes. This difference is important. The core kernel address space is a direct map of physical memory; it can be handled very efficiently in the processor's page table. Indeed, on some processors, a single page table entry covers the entire kernel. Space obtained from `vmalloc`, instead, uses one page table entry per memory page. A greater number of page table entries mean more lookups, and more translation buffer misses.

One estimate is that the slowdown can be as much as 5%. Given this problem, why not load modules into the regular kernel memory space? Module code requires a contiguous address space. Since the standard kernel space is a direct map of physical memory, contiguous address spaces must also be contiguous in physical memory. Once the system has been running for a while, finding even two physically contiguous pages can be a challenge; finding enough to load a large module can be almost impossible. Modules also seem to have endemic problems with race conditions - it is possible, for example, for the kernel to attempt to access a newly-loaded module before it is fully initialized. Modules can also, in some situations, be removed while still in use. Such occurrences are obviously quite rare, but they can be catastrophic when they happen. The race conditions can be fixed with enough work, but that may require changing some fundamental kernel interfaces. In general, dealing with loadable modules is not an easy task; as one kernel hacker told us in a private message: "Doing live surgery on the kernel is never going to be pretty."

These installation instructions assume

Commands are Unix-compatible.
The source path is `/usr/src`.
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account "`root`".
Latest Kernel version number is `2.4.18`

The procedures given in this chapter are likely to work on all Linux platforms, but we have only tested it on OpenNA Linux and Red Hat Linux.

Packages

The following is based on information listed by the Linux Kernel Archives as of 2002/06/04. Please check http://www.kernel.org/ regularly for the latest status. We chose to install from source because it provides the facility to fine tune the installation.

Source code is available from:
Kernel Homepage: http://www.kernel.org/
Kernel `FTP` Site: `204.152.189.116`
You must be sure to download: `linux-2.4.18.tar.gz`

Prerequisites

Depending on whether you want a firewall, user quota support with your system or if you have a `SCSI/RAID` controller, the Linux `Kernel` requires that the listed software below be already installed on your system to be able to compile successfully. If this is not the case, you must install them from your Linux CD-ROM or source archive files. Please make sure you have all of these programs installed on your system before proceeding with this chapter.

- ✓ `iptables` package, is the new secure and more powerful program used by Linux to set up firewalls as well as `IP` masquerading on your system. Install this package if you want to support Firewalls on your server.

- ✓ `quota` package, is a system administration tool for monitoring and limiting users' and/or groups' disk usage, per file system. Install this package if you want a tool to control the size of user's directories on your server.

- ✓ `mkinitrd` package, creates filesystem images for use as initial ramdisk (`initrd`) images. These ramdisk images are often used to preload the block device modules (`SCSI` or `RAID`) needed to access the root filesystem. Install this package if you have a `SCSI` or `RAID` system where the Kernel is compiled as a Modularized Kernel.

- ✓ `mkbootdisk` package, creates a standalone boot floppy disk for booting the running system. Install this package only if you have a Modularized Kernel installed on your system. This package is not needed for Monolithic Kernel.

- ✓ The `dosfstools` package includes the `mkdosfs` and `dosfsck` utilities, which make and check MS-DOS FAT filesystems on hard drives or on floppies. You only need to install this package on Modularized Kernel.

NOTE: For more information on `Iptables` Netfilter Firewall configuration or `quota` software, see the related chapters later in this book.

Pristine source

If you don't use the RPM package to install the kernel, it will be difficult for you to locate all the files installed onto the system if you want to update your kernel in the future. To solve this problem, it's a good idea to make a list of files on the system before you install the kernel, and then one afterwards, and then compare them using the **diff** utility to find out what files were placed where.

- Simply run the following command before installing the kernel:
```
[root@deep root]# find /* > Kernel1
```

- And the following one after you install the kernel:
```
[root@deep root]# find /* > Kernel2
```

- Then use this command to get a list of what changed:
```
[root@deep root]# diff Kernel1 Kernel2 > Kernel-Installed
```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new kernel. In our example above, we use the /root directory of the system to store all the generated file lists.

Making an emergency boot floppy

The first step before going into the configuration and compilation of our new kernel is to create an emergency boot floppy in case something goes wrong during the build of your new Linux Kernel. Here we create the boot floppy for a modularized kernel since the kernel that is presently installed on our system should be a modularized kernel. We'll see later that a method for creating the boot disk for mololithic kernel exists and is different from what we use here. To create the emergency boot floppy for a modularized kernel, follow these steps.

Step1
We have to find the present modularized kernel version used on our system. We need this information to be able to create our emergency boot floppy disk.

- To know which kernel version is running on your system, use the command:
```
[root@dev /]# uname -a
Linux dev 2.4.18-3 #1 Thu Apr 18 07:37:53 EDT 2002 i686 unknown
```

From the above command, we know now that our kernel version is 2.4.18-3. Therefore we will use this information in the next step to create the boot disk.

Step2
Once we know which kernel version we are currently running, we can use the command below to create the boot disk.

- Put a floppy in your system and execute the following command as root:
```
[root@deep /]# mkbootdisk --device /dev/fd0H1440 2.4.18-3
Insert a disk in /dev/fd0. Any information on the disk will be lost.
Press <Enter> to continue or ^C to abort:
```

NOTE: In this example, the current kernel on our system is version `2.4.18-3` and this is why we use "`2.4.18-3`" here. If you kernel version is different from what we use here, you just have to change the example version number for the one that you have according to the result returned by the "**`uname -a`**" command.

Following these guidelines, you will now have a boot floppy with a known working kernel in case of problems with the upgrade. I recommend rebooting the system with the floppy to make sure that the floppy works correctly before continuing.

Preparing the Kernel for the installation

We have to copy the new kernel tar archive to the appropriate location on our server `/usr/src` and then we remove the old kernel from our system before installing a new one. Removing the old kernel will not freeze your computer until you try to reboot it before installing the new one because the Linux kernel resides in memory.

Step 1
We must copy the archive file of the kernel to the `/usr/src` directory and move to this directory.

- To copy the tar archive of the Linux kernel to the `/usr/src` directory, use the command:
 `[root@deep /]#` **`cp linux-version.tar.gz /usr/src/`**

- To move to the `/usr/src` directory, use the following command:
 `[root@deep /]#` **`cd /usr/src/`**

Step 2
Depending on how the Linux Kernel has been previously installed on your system, there are two ways to uninstall it, these are shown below.

If you already have installed a Linux kernel with a tar archive before
These steps are required ONLY if you have previously installed a Linux kernel from a tar archive. If it is a fresh, first install of the kernel, then uninstall the `kernel-headers-version.i386.rpm`, `kernel-version.i386.rpm` packages that are on your system.

- Move to the `/usr/src` directory if you are not already in it with the following command:
 `[root@deep /]#` **`cd /usr/src/`**

- Remove the Linux symbolic link with the following command:
 `[root@deep src]#` **`rm -f linux`**

- Remove the Linux kernel headers directory with the following command:
 `[root@deep src]#` **`rm -rf linux-2.4.x/`**

- Remove the Linux kernel with the following command:
 `[root@deep src]#` **`rm -f /boot/vmlinuz-2.4.x`**

- Remove the Linux `System.map` file with the following command:
 `[root@deep src]#` **`rm -f /boot/System.map-2.4.x`**

- Remove the Linux kernel modules directory (if available) with the following command:
 `[root@deep src]#` **`rm -rf /lib/modules/2.4.x/`**

NOTE: Removing the old kernel modules is only required if you have installed a modularized kernel version before. If the modules directory doesn't exist under the `/lib/modules` directory, it's because your old kernel version is not a modularized kernel.

If the original kernel's `RPM` packages are installed on your system

If the original kernel `RPM` packages are installed on your system, instead of the Linux kernel tar archive, which they would be if you have just finished installing your new Linux system, or have previously used an `RPM` package to upgrade your system, then use the following command to uninstall the Linux kernel:

- Verify which kernel `RPM` packages are installed on your server with the command:
```
[root@deep src]# rpm -qa | grep kernel
kernel-2.4.18-3
```

- Also, verify if Kernel header package is installed on your server with the command:
```
[root@deep src]# rpm -q glibc-kernheaders
glibc-kernheaders-2.4-7.14
```

The above command shows us that `kernel` and `glibc-kernheaders` are the only kernel `RPM` packages installed on our system. We uninstall them as show below.

- To uninstall the linux kernel `RPM`, use the following command:
```
[root@deep src]# rpm -e --nodeps kernel glibc-kernheaders
```

NOTE: If you receive an error message like: cannot remove `/lib/modules/2.4.x` directory, directory not empty, then remove the directory manually with command like: `rm –rf /lib/modules/2.4.x/` form your system. This directory is related to the old kernel and it is not required for the new kernel we want to install.

Step 3

Once we have uninstalled the old kernel and our new kernel tar archive has been copied to the `/usr/src` directory, we must uncompress it and remove the tar archive (`linux-version.tar.gz`) from the system to conserve disk space.

- To uncompress the kernel, use the following command:
```
[root@deep src]# tar xzpf linux-version.tar.gz
```

- To remove the kernel tar archive from the system, use the following command:
```
[root@deep src]# rm -f linux-version.tar.gz
```

WARNING: If kernel compilation is something new for you, then it is recommended to keep the kernel tar archive (`linux-version.tar.gz`) until the end of the installation. This way, if you make a mistake during compilation, you have the source available to try again.

Applying the Grsecurity kernel patch

Grsecurity is a single patch file for the newest stable versions of Linux kernel that is an attempt to greatly improve the security of a Linux system. It mainly accomplishes this by physically patching the Linux kernel to make processes more restricted.

Many other projects like Grsecurity exist on the Internet. There are some well know like the RSBAC (www.rsbac.org), LIDS (www.lids.org), SELinux (www.nsa.gov/selinux/), and OpenWall (www.openwall.com/linux/), projects, all of which fulfills only one part of a complete security system.

What makes Grsecurity so different and better than these other projects is mainly because Grsecurity provides greatly needed additional security to Linux systems. In other words, it covers all features of the other projects and adds additional security features that the other projects do NOT cover. The types of added Grsecurity security are categorized as:

1. Complete Buffer Overflow Exploitation Protection.
2. File system Race Protection.
3. System Auditing.
4. Change-root Protections.
5. Portscan and OS Fingerprinting Protection.
6. Restricted Users.
7. Configurability Options.
8. Access Control.

Grsecurity patch may change from version to version, and some may contain various other security features. It is important to use the Grsecurity patch that corresponds to the Linux Kernel version that you compile on your server. If you compile kernel version 2.4.18, you have to download Grsecurity patch for kernel version 2.4.18, etc.

WARNING: When applying the Grsecurity patch to the Kernel, a new "Grsecurity" configuration section will be added at the end of your Linux Kernel configuration allowing you to configure and enable the security features that you want.

Obtaining and Installing Grsecurity

To obtain the Grsecurity patch suitable for your Linux Kernel, simply follow the "download" link on the Grsecurity home page, and download the file listed there. Grsecurity patch is available directly via http://www.grsecurity.org/. Remember that Grsecurity is specific to one kernel version, and as of this writing that version is 2.4.18.

- To apply the Grsecurity patch to the Linux kernel, use the commands:

```
[root@deep /]# cp grsecurity-1.9.4-2.4.18.patch /usr/src/
[root@deep /]# cd /usr/src/linux/
[root@deep linux]# patch -p1 < ../grsecurity-1.9.4-2.4.18.patch
[root@deep linux]# cd ../
[root@deep src]# rm -f grsecurity-1.9.4-2.4.18.patch
```

The step of patching your new kernel with Grsecurity patch is completed. Now follow the rest of this Kernel installation guide to build the Linux kernel and reboot your system. Configuration relating to Grsecurity will appears at the end of your Kernel configuration.

Tuning the Kernel

Ok, the old kernel has been uninstalled from our system; we have copied the new one to its appropriate location, uncompressed it, and added the `Grsecurity` patch (if wanted). Now, we must tune our new kernel to the maximum of its capabilities. All optimizations shown below are just increases of the default kernel parameters.

- Edit the **`sem.h`** file (`vi +66 /usr/src/linux/include/linux/sem.h`) and change the following parameter:

  ```
  #define SEMMNI 128 /* <= IPCMNI max # of semaphore identifiers */
  ```

 To read:

  ```
  #define SEMMNI 512 /* <= IPCMNI max # of semaphore identifiers */
  ```

- Edit the **`limits.h`** file (`vi /usr/src/linux/include/linux/limits.h`) and change the following parameters:

  ```
  #define NR_OPEN               1024
  ```

 To read:

  ```
  #define NR_OPEN               8192


  #define OPEN_MAX         256   /* # open files a process may have */
  ```

 To read:

  ```
  #define OPEN_MAX        8192   /* # open files a process may have */
  ```

- Edit the **`posix_types.h`** file (`vi +25 /usr/src/linux/include/linux/posix_types.h`) and change the following parameter:

  ```
  #define __FD_SETSIZE   1024
  ```

 To read:

  ```
  #define __FD_SETSIZE   8192
  ```

Finally, we must instruct the kernel to fit our specific `CPU` architecture and optimization flags. Depending on your `CPU` architecture and optimization flags, this step will improve the performance of the kernel. As an example with a PII 400MHz the BogoMIPS will become 799.54 instead of the default number of 400.00.

Take note that it is not because BogoMIPS show you a number of 799.54 for a 400MHz CPU that your processor runs at this speed now. The BogoMIPS result can just be considered as a benchmark since it is a meaningless benchmark measurement.

- Edit the **Makefile** file (`vi +20 /usr/src/linux/Makefile`) and change the line:

  ```
  HOSTCFLAGS = -Wall -Wstrict-prototypes -O2 -fomit-frame-pointer
  ```

 To read:

  ```
  HOSTCFLAGS = -Wall -Wstrict-prototypes -O2 -march=i686 -funroll-loops -
  fomit-frame- pointer
  ```

- Edit the **Makefile** file (`vi +91 /usr/src/linux/Makefile`) and change the line:

  ```
  CFLAGS := $(CPPFLAGS) -Wall -Wstrict-prototypes -O2 -fomit-frame-pointer
  -fno-strict-aliasing
  ```

 To read:

  ```
  CFLAGS := $(CPPFLAGS) -Wall -Wstrict-prototypes -O2 -march=i686 -funroll-
  loops -fomit-frame-pointer -fno-strict-aliasing
  ```

WARNING: In the last example, we optimize the code for an i686 `CPU` architecture, if you have a different processor, you'll have to adjust the "`-march=i686`" options for your specific processor.

Never compile the Kernel with optimization code number superior to "`-O2`", this do nothing more and should produce an unstable kernel in some cases. Therefore use "`-O2`" at the maximum optimization number but never something superior to it.

Cleaning up the Kernel

It is important to make sure that your `/usr/include/asm`, and `/usr/include/linux` subdirectories are just symlinks to the kernel sources.

Step 1
The `asm` and `linux` subdirectories are soft links to the real include kernel source header directories needed for our Linux architecture, for example `/usr/src/linux/include/asm-i386` for `asm`.

- To symlink the `asm`, and `linux` subdirectories to the kernel sources, type the following commands on your terminal:

  ```
  [root@deep src]# cd /usr/include/
  [root@deep include]# rm -f asm linux
  [root@deep include]# ln -s /usr/src/linux/include/asm-i386 asm
  [root@deep include]# ln -s /usr/src/linux/include/linux linux
  ```

This is a very important part of the configuration: we remove the `asm`, and `linux` directories under `/usr/include` then create new links that point to the same name directories under the new kernel source version directory. The `/usr/include` directory contains important header files needed by your kernel and programs to be able to compile on your system.

WARNING: If the previously installed kernel in your system was made by `RPM` packages, then the `asm` and `linux` soft links will not exist since the uninstall of `kernel-headers` `RPM` package removes them automatically for you. Don't forget to create them.

Step 2
Make sure you have no stale `.o` files or dependencies lying around.

- To be sure that we have no stale `.o` files or dependencies lying around, type the following commands on your terminal:

```
[root@deep include]# cd /usr/src/linux/
[root@deep linux]# make mrproper
```

WARNING: These two steps simply clean up anything that might have accidentally been left in the source tree by the development team.

You should now have the source correctly installed. You can configure the kernel in one of three ways. The first method is to use the **`make config`** command. It provides you with a text-based interface for answering all the configuration options. You are prompted for all the options you need to set up your kernel.

The second method is to use the **`make menuconfig`** command, which provides all the kernel options in an easy-to-use menu. The third is to use the **`make xconfig`** command (only available if the graphical interface of Linux is installed on the system), which provides a full graphical interface to all the kernel options.

Step 3
For configuration in this guide, we will use the **`make config`** command because we have not installed the XFree86 Window Interface on our Linux server or the necessary packages to use the **`make menuconfig`** command.

- Type the following commands on your terminal to load the kernel configuration:

```
[root@deep /]# cd /usr/src/linux/ (if you are not already in this directory).
[root@deep linux]# make config
rm -f include/asm
( cd include ; ln -sf asm-i386 asm)
/bin/sh scripts/Configure arch/i386/config.in
#
# Using defaults found in arch/i386/defconfig
#
```

Configuring the Kernel

As soon as you enter **`make config`** at the prompt as described in the previous step, a list of kernel configurable options will be displayed for you to choose to configure the kernel, you must indicate what features and devices drivers you want to include on your system and select how to include support for specific devices. Typically, for each configuration option, you have to respond with one of the following choices:

`[y]` To compile into the kernel and always be loaded.
`[m]` To use a module for that feature and load that segment of code on demand.
`[n]` To skip and excludes the support for that specific device from the kernel.

> **WARNING:** It is important to note that an **`[n]`** or **`[y]`** means the default choice. If a device does not have a modular device driver or you have not compiled the Kernel as a Modularized Kernel, you will not see the **`[m]`** option. Some time an **`[?]`** option will appear in the choices. This mean that you can get more information about the feature when you type the **`? + ENTER`** key. Choosing the **`[?]`** help option opens another terminal describing the option.

A new Linux kernel is very specific to our computer hardware since we have to choose the right drivers as well as features that we need to include and compile into the Kernel code. This implies a good understanding and knowledge of your computer hardware. It is simply inconceivable to build a Linux system if you don't know what hardware your computer has, especially if you spend money to buy a computer and then take time to configure it. Therefore we assume that you know all of your hardware and be aware that during the kernel configuration, you will be asked to answer some important questions related to your specific computer hardware.

Be prepared to answer the following questions:

1. What type of processor do you have on your computer (i.e. Pentium III, AMD)?
2. How many processor do you have on your computer (i.e. 1, 2, 3)?
3. What kind of hard drive do you have on your computer (i.e. IDE, SCSI) ?
4. How many hard drives do you have on your computer? (i.e. want to make RAID)?
5. How much memories (RAM) do you have on your computer (i.e. 512 MB RAM)?
6. Do you have a network card? If so, who made it and what model is it?
7. Do you have a SCSI adapter? If so, who made it and what model is it?
8. Do you have a RAID system? If so, who made it and what model is it?
9. What type of mouse do you have (eg, PS/2, Microsoft, Logitech)?
10. If you have a serial mouse, what COM port is it connected to (eg, COM1)?
11. What is the make and model of your video card?

All of the above questions are very important and if you don't know the answers for all of them, then it is recommended to get the information before going into Linux kernel configuration, compilation and installation. If you have all of the information, then you can read the rest of this guide.

Monolithic kernel configuration

As we know now, they are two possible different configurations for the kernel. The first is called a `monolithic kernel` the second is called a `modularized kernel`. Below we begin by showing you the configuration of a `monolithic kernel` which is to compile the required code and drivers directly into the kernel by answering the different kernel questions only by **`yes`** or **`no`**. Don't forget to only compile code that you need and use.

A new kernel is very specific to your computer hardware, in the `monolithic kernel` configuration part below; we have the following hardware for our example. Of course you must change them to fit whatever components you have in your system.

1 Pentium II 400 MHz (i686) processor
1 SCSI Motherboard
1 SCSI Hard Disk
1 SCSI Controler Adaptec AIC 7xxx
1 CD-ROM ATAPI IDE
1 Floppy Disk
2 Ethernet Cards Intel EtherExpressPro 10/100
1 Mouse PS/2

If you don't want some of the options listed in the `monolithic kernel` configuration that I enable by default, answer **`n`** (for no) instead of **`y`** (for yes) to the related questions. If you want some other options that I disable, then answer **`y`** instead of **`n`**. Finally, the procedure of building a new kernel is quite long, therefore I recommend you to take your time. Some coffees and cigarettes will surely be welcome during these steps.

```
# Using defaults found in arch/i386/defconfig
#
*
* Code maturity level options
* Prompt for development and/or incomplete code/drivers (CONFIG_EXPERIMENTAL)
[N/y/?] Press Enter
```

This option relates to features or drivers that are currently considered to be in the alpha-test phase and for a production system, it is highly recommended to answer by **`N`** to this question. Just hit **`[Enter]`** to approve the default choice, which is **`NO`** (Do not prompt for development and/or incomplete code/drivers).

```
*
* Loadable module support
* Enable loadable module support (CONFIG_MODULES) [Y/n/?] n
```

This option is very important since it asks us if we want to enable loadable module support into the Kernel. Remember that our goal in this part of the guide is to build a Monolithic Linux Kernel where all drivers and features are directly integrated and compiled into the Kernel code, therefore our answer to this question must be **`No`** (Do not enable loadable module support). This means that we have to enter **`n`** as the answer to this question to change the default value, which is **`Y`** for Yes.

```
*
* Processor type and features
* Processor family (386, 486, 586/K5/5x86/6x86/6x86MX, Pentium-Classic,
Pentium-MMX, Pentium-Pro/Celeron/Pentium-II, Pentium-III/Celeron(Coppermine),
Pentium-4, K6/K6-II/K6-III, Athlon/Duron/K7, Crusoe, Winchip-C6, Winchip-2,
Winchip-2A/Winchip-3, CyrixIII/C3) [Pentium-III/Celeron(Coppermine)] Pentium-4
```

This option asks you what type of processor you have on your computer and the default choice is in brackets `[Pentium-III/Celeron(Coppermine)]`. Therefore, if you processor is not a Pentium-III/Celerom(Coppermine) model, you have to enter your processor model here. The available choices are listed in the question. In the above example, I changed the default value `[Pentium-III/Celeron(Coppermine)]` and chose a `Pentium-4` model.

```
Toshiba Laptop support (CONFIG_TOSHIBA) [N/y/?] Press Enter
```

This option asks you if you want to add a driver support for Toshiba Laptop. If you intend to run this kernel on a Toshiba portable, then you have to answer `Yes` to this question and change the default value which is `No`. In our example, we keep the default value of `No` by pressing the `[Enter]` key.

```
Dell Inspiron 8000 support (CONFIG_I8K) [N/y/?] Press Enter
```

As for the previous option, this one asks you if you want to add a driver support for Dell Inspiron 8000 Laptop. If you intend to run this kernel on a Dell Inspiron 8000, then you have to answer `Yes` to this question and change the default value which is `No`. In our example, we keep the default value of `No` by pressing the `[Enter]` key again.

```
/dev/cpu/microcode - Intel IA32 CPU microcode support (CONFIG_MICROCODE)
[N/y/?] Press Enter
```

This option allows you to update the microcode on Intel processors in the IA32 family, e.g. Pentium Pro, Pentium II, Pentium III, Pentium 4, Xeon etc. If you say `Y` here and change the default value of `N`, then you'll need to say `Y` also to "`/dev file system support`" in the 'File systems' section below. In most situations, this option is simply not needed and you can safety keep the default setting of `N` by pressing the `[Enter]` key.

```
/dev/cpu/*/msr - Model-specific register support (CONFIG_X86_MSR) [N/y/?] y
```

This option allows you to enable a device that gives privileged processes access to the x86 Model-Specific Registers (MSRs). On multi-processor the MSR accesses are directed to a specific CPU. It is a good idea to answer `Y` to this option to enable it, even if you only have one processor on your system. Therefore answer `Y` to this question.

```
/dev/cpu/*/cpuid - CPU information support (CONFIG_X86_CPUID) [N/y/?] y
```

This option allows you to enable a device that gives processes access to the x86 CPUID instructions to be executed on a specific processor. As for the previous option, it is a good idea to change the default value of `N` to `Y`.

```
High Memory Support (off, 4GB, 64GB) [off] Press Enter
```

This option allows you to be able to use up to 64 Gigabytes of physical memory on x86 systems. Usually, and it's true for most of us, we have a system that will never run with more than 1 Gigabyte total physical RAM and if this is you case, then answer "`off`" here (default choice and suitable for most users). In other part if you have a system that has between 1 and 4 Gigabytes physical RAM, then answer "`4GB`" here. If more than 4 Gigabytes is used then answer "`64GB`" here. In our example, we will configure the kernel to use a maximum of 1 Gigabyte total physical RAM by pressing the `[Enter]` key to accept the default choice "`off`".

```
Math emulation (CONFIG_MATH_EMULATION) [N/y/?] Press Enter
```
This option allows Linux to emulate a math coprocessor (a feature used for floating point operations). In general, only old 486SX and 386 processors do not have a a math coprocessor built in. All modern Pentium I, II, III, IV and later, AMD, Cyrix, etc have a math coprocessor built in and do not require this option to be turned on. If you have a very old 486SX or 386 processor in your computer, then you will have to change the default value of `N` here to become `Y`, but in general, everyone will keep the default value of `N` here since they have a math coprocessor built in their processor chip.

```
MTRR (Memory Type Range Register) support (CONFIG_MTRR) [N/y/?] Press Enter
```
This option on Intel family processors (Pentium Pro, Pentium II and later) allows the Memory Type Range Registers (MTRRs) to be used to control processor access to memory ranges. This is most useful if you have XFree86 graphical interface installed on your computer or have more than 1 processor on your system. Changing the default value of `N` here to become `Y` on a Linux system where a graphical interface or multiple processors are present will increase the performance of the system. If you don't use a graphical interface, or do not have more than one processor installed on you system, you must keep the default value of `N` by pressing the `[Enter]` key. Finally, it is important to note that this option is valid only if you have Intel family processors (Pentium Pro, Pentium II and later) installed on your computer. It doesn't work with AMD or Cyrix processors.

```
Symmetric multi-processing support (CONFIG_SMP) [Y/n/?] n
```
This option enables support for systems with more than one CPU. If you have a system with only one CPU, like most personal computers, say `N`. If you have a system with more than one CPU, say `Y`. In our example, we only have one processor installed on our computer and will change the default value of `Y` to become `N`.

```
Local APIC support on uniprocessors (CONFIG_X86_UP_APIC) [N/y/?] (NEW) Press
Enter
```
This option appears only if you have answered `N` to the previous option and will let you enable support for local APIC on system with single processor. Local APIC (Advanced Programmable Interrupt Controller) is an integrated interrupt controller in the CPU that supports CPU-generated self-interrupts (timer, performance counters), and the NMI watchdog, which detects hard lockups on the server. On systems with single processor, you may have to answer `Y` here, but on systems with multiple processors, you don't have to answer this question since the kernel will automatically enable it. Usually we keep the default choice of `N` here since the feature is available only on modern Pentium processors like the IV family. Pentium I, II, and III don’t support this option, as well as AMD and Cyrix processors.

```
*
* General setup
* Networking support (CONFIG_NET) [Y/n/?] Press Enter
```
This option is very important and must be set to `Y` all the time, which is the default value. It allows your Linux system to support and use networking. We use the default value of `Y` by pressing the `[Enter]` key.

```
PCI support (CONFIG_PCI) [Y/n/?] Press Enter
```
This option allows us to enable or disable PCI support on Linux. In most cases, we have to say `Y` here. PCI's are the white slots on your motherboard where you can add network cards, video cards, etc. Since most of us use a PC and have this kind of slot available, it is important to say `Y` here by pressing the `[Enter]` key to use the default choice.

`PCI access mode (BIOS, Direct, Any) [Any]` **Press Enter**
On PCI systems, the BIOS can be used to detect the PCI devices and determine their configuration. However, some old PCI motherboards have BIOS bugs and may crash if this is done. Also, some embedded PCI-based systems don't have any BIOS at all. Linux can also try to detect the PCI hardware directly without using the BIOS. With this option, you can specify how Linux should detect the PCI devices. If you choose "`BIOS`", the BIOS will be used, if you choose "`Direct`", the BIOS won't be used, and if you choose "`Any`", the kernel will try the direct access method and falls back to the BIOS if that doesn't work. If unsure, go with the default, which is "`Any`" by pressing the `[Enter]` key.

`PCI device name database (CONFIG_PCI_NAMES) [Y/n/?]` **n**
This option lets you enable a feature that allows the Kernel to contain a database of all known `PCI` device names via different files under the `/proc` filesystem and make the information comprehensible to the user or disable this feature and get device ID numbers instead of names. Disabling this feature will save you about 80KB of kernel image size and will make the kernel smaller in size, which is a good thing for increased performance. If you disable this feature, the kernel will still run as usual, but will show you device ID numbers instead of names. Therefore, we will change the default value of `Y` (get `PCI` device by names) to become `N` (get `PCI` device by ID numbers) and will save 80KB of Kernel image size.

`EISA support (CONFIG_EISA) [N/y/?]` **Press Enter**
This option allows you to enable support for the **E**xtended **I**ndustry **S**tandard **A**rchitecture (`EISA`) bus. `EISA` is now obsolete by the `PCI` bus and you may enable this option only if you use some older `ISA` card on your system. Since most of us don't have these types of card, we can safety keep the default value of `N` here by pressing the `[Enter]` key.

`MCA support (CONFIG_MCA) [N/y/?]` **Press Enter**
This option allows us to enable `MCA` support with the kernel. `MCA` (**M**icro**C**hannel **A**rchitecture) is found in some IBM PS/2 machines and laptops. It is a bus system similar to `PCI` or `ISA`. It is rare that we have this kind of bus on our system and we can safety keep the default value of `N` (do not support `MCA` on this kernel) by pressing the `[Enter]` key again.

`Support for hot-pluggable devices (CONFIG_HOTPLUG) [Y/n/?]` **n**
This option is often required only on laptop computer where you have a `PCMCIA` or PC-cards that you can plug or unplug at any time. If the kernel that you compile is made to run on a standard computer (not laptop, portable), then you have to say `N` here by changing the default option of `Y` to become `N`.

`System V IPC (CONFIG_SYSVIPC) [Y/n/?]` **Press Enter**
This option allows us to enable `IPC` on Linux. `IPC` (**I**nter **P**rocess **C**ommunication) is a suite of library functions and system calls which let processes (running programs) synchronize and exchange information. It is generally considered to be a good thing, and some programs won't run unless you say `Y` here. Therefore, we keep the default setting of `Y` by pressing the `[Enter]` key.

`BSD Process Accounting (CONFIG_BSD_PROCESS_ACCT) [N/y/?]` **Press Enter**
This option allows us to enable a user level programs, which will be able to instruct the kernel (via a special system call) to write process accounting information to a file. It is not vital or even really necessary to enable this kind of option to get a working Linux kernel. Therefore, we keep the default value of `N` here by pressing the `[Enter]` key.

`Sysctl support (CONFIG_SYSCTL) [Y/n/?]` **Press Enter**
This option is very important and especially with Linux kernel 2.4.x generation. It is the feature that allows us to dynamically change certain kernel parameters and variables on the fly through the `/proc` filesystem with the use of the `/etc/sysctl.conf` file. We keep the default value of `Y` by pressing the `[Enter]` key.

`Kernel core (/proc/kcore) format (ELF, A.OUT) [ELF]` **Press Enter**
This option allows a file under the `/proc` filesystem, which is called "`kcore`" to contain the Kernel core image in two different formats, which are `ELF` or `A.OUT`. For newer Linux systems, `ELF` (**E**xecutable and **L**inkable **F**ormat) is highly recommended and is the default option that we choose by pressing the `[Enter]` key. `A.OUT` (**A**ssembler.**OUT**put) was the old format method used, but is now really obsolete.

`Kernel support for a.out binaries (CONFIG_BINFMT_AOUT) [Y/n/?]` **n**
This option is very important and allows us to provide support for `A.OUT`. `A.OUT` (**A**ssembler.`OUT`put) is a set of formats for libraries and executables used in the earliest versions of UNIX and, as we said before, it is now really obsolete and was replaced with the `ELF` format. To be sure none of our programs will use this older executable format, we will change the default value of `Y` to become `N`.

`Kernel support for ELF binaries (CONFIG_BINFMT_ELF) [Y/n/?]` **Press Enter**
Here, it is important to answer `Y` to this question since this binary format is the one used now on modern Linux systems. `ELF` (**E**xecutable and **L**inkable **F**ormat) is a format for libraries and executables used across different architectures and operating systems. Many new executables are distributed solely in `ELF` format and you definitely want to say `Y` to this option by pressing the `[Enter]` key.

`Kernel support for MISC binaries (CONFIG_BINFMT_MISC) [Y/n/?]` **Press Enter**
This option enables wrapper-driven binary formats into the kernel. This feature is required by programs that need an interpreter to run, like Java, Python or Emacs-Lisp. Since we're sure to use one of these types of programs, it is safe to accept the default value of `Y` by pressing the `[Enter]` key.

`Power Management support (CONFIG_PM) [Y/n/?]` **n**
This option allows Power Management Support for your computer. With this feature parts of your computer are shut off or put into a power conserving "sleep" mode if they are not being used. In general it is good to enable this option on portable computer (Laptop) only. Note that, even if you say `N` here, Linux, on the x86 architecture, will issue the `hlt` instruction if nothing is to be done, thereby sending the processor to sleep and saving power. Therefore our choice will be `N` here.

```
*
* Memory Technology Devices (MTD)
* Memory Technology Device (MTD) support (CONFIG_MTD) [N/y/?] Press Enter
```
This option enables `MTD` (**M**emory **T**echnology **D**evices) on your computer. `MTD` are flash, RAM and similar chips, often used for solid-state file systems on embedded devices. If you have some of these devices in your system, then answer `Y` to the question. In most cases, the default choice of `N` is recommended.

```
*
* Parallel port support
* Parallel port support (CONFIG_PARPORT) [N/y/?] Press Enter
```
If you want to use devices connected to your machine's parallel port like a printer, zip drive, etc, then you need to say `Y` here otherwise the default value `N` is recommended.

```
*
* Plug and Play configuration
* Plug and Play support (CONFIG_PNP) [Y/n/?] n
```

Plug a**n**d **P**lay (PnP) is a standard for peripherals which allows those peripherals to be configured by software. If you answer to this question by Y, then Linux will be able to configure your Plug and Play devices. Under Linux, we really don't need to enable PNP support and our choice will be N here.

```
*
* Block devices
* Normal PC floppy disk support (CONFIG_BLK_DEV_FD) [Y/n/?] Press Enter
```

This option allows us to use the floppy disk drive(s) in our PC under Linux. Since everyone has and usually needs a floppy disk in their computer, the answer to this question will be Y. If you run a Linux server in highly secure environment, you could answer to this question by N since we never use floppy disk on this type of system.

```
XT hard disk support (CONFIG_BLK_DEV_XD) [N/y/?] Press Enter
```

This option allows us to enable the very old 8 bit hard disk controllers used in the IBM XT computer and since it's pretty unlikely that you have one of these then you must answer to this question by N. Therefore we simply press [Enter] because the default answer to this question is N.

```
Compaq SMART2 support (CONFIG_BLK_CPQ_DA) [N/y/?] Press Enter
```

This option allows us to enable support for the Compaq Smart Array controllers. If you use these kinds of boards in your system, then you should say Y here, otherwise the default value of N is recommended.

```
Compaq Smart Array 5xxx support (CONFIG_BLK_CPQ_CISS_DA) [N/y/?] Press Enter
```

This option allows us to enable support for the Compaq Smart Array 5xxx controllers. If you use these kinds of boards in your system, then you should say Y here, otherwise the default value of N is recommended.

```
Mylex DAC960/DAC1100 PCI RAID Controller support (CONFIG_BLK_DEV_DAC960)
[N/y/?] Press Enter
```

This option enables support for the Mylex DAC960, AcceleRAID, and eXtremeRAID PCI RAID controllers. If you use these kinds of boards in your system, then you should say Y here, otherwise the default value of N is recommended.

```
Loopback device support (CONFIG_BLK_DEV_LOOP) [N/y/?] Press Enter
```

This option is a little different from other kernel options and if you enable it, you will be able to use a regular file as a block device that let you have access to some special advanced features and possibilities under Linux. The default value for this option will be ok for most of us and only advanced users or user that know why they need these features will enable the "Loopback device support" option. For SCSI systems with a modularized kernel, it is important to say Y here since SCSI drivers will use this device.

```
Network block device support (CONFIG_BLK_DEV_NBD) [N/y/?] Press Enter
```

This option enables another advanced feature under Linux, the possibility for your system to be a client for network block devices. The default value for this option would be ok for most of us and only advanced users or user that know why they need this feature will enable it.

RAM disk support (CONFIG_BLK_DEV_RAM) [N/y/?] **Press Enter**
This option will allow you to use a portion of your RAM memory as a block device, so that you can make file systems on it, read and write to it and do all the other things that you can do with normal block devices (such as hard drives). It is usually used to load and store a copy of a minimal root file system off of a floppy into RAM during the initial install of Linux. Again, most normal users won't need the RAM disk functionality and will answer N to this question. For SCSI systems with a modularized kernel, it is important to say Y here since SCSI drivers will use this feature.

```
*
* Multi-device support (RAID and LVM)
* Multiple devices driver support (RAID and LVM) (CONFIG_MD) [N/y/?] Press
Enter
```
This option is required only for RAID and **l**ogical **v**olume **m**anagement (LVM). If you use them, then change the default value of N to become Y.

```
*
* Networking options
* Packet socket (CONFIG_PACKET) [Y/n/?] Press Enter
```
This option allows you to enable applications, which communicate directly with network devices without an intermediate network protocol implemented in the kernel like the tcpdump program. It is a good idea to enable this feature for most of us.

Packet socket: mmapped IO (CONFIG_PACKET_MMAP) [N/y/?] **y**
This option allows packet protocol driver to use an IO (**I**nput/**O**utput) mechanism that results in faster communication. Say Y here.

Kernel/User netlink socket (CONFIG_NETLINK) [N/y/?] **y**
This option allows us to enable two-way communication between the kernel and user processes. Say Y here.

Routing messages (CONFIG_RTNETLINK) [N/y/?] (NEW) **y**
If we have said Y to the previous option, we must say Y here too or the previous option will not work. Therefore our choice is Y.

Netlink device emulation (CONFIG_NETLINK_DEV) [N/y/?] (NEW) **y**
This option is a backward compatibility option, and we have to choose Y for now.

Network packet filtering (replaces ipchains) (CONFIG_NETFILTER) [N/y/?] **y**
This option enables support for a packet filter firewall on your system (Netfilter). It is very important to answer to this question by Y if you want to support firewall and IPTables on your computer. If you answer N to this question, then the firewalling features will not be available, even if your have the IPTables software installed on your system.

Network packet filtering debugging (CONFIG_NETFILTER_DEBUG) [N/y/?] (NEW) **y**
This option turns on support for debugging the netfilter code. It is a good idea to enable it.

Socket Filtering (CONFIG_FILTER) [N/y/?] **Press Enter**
This option allows us to enable Linux Socket Filter, a feature needed by PPP packet filtering in general. Therefore you only need to say Y here if you want to use PPP packet filtering on your system. Since we use a network card to get a connection to the Internet and not PPP (modem link), we keep the default value of N here.

`Unix domain sockets (CONFIG_UNIX) [Y/n/?]` **Press Enter**
This option is very important and must always be set to `Y`. It allows us to include support for Unix domain sockets; sockets are the standard Unix mechanism for establishing and accessing network connections. It is vital to say `Y` to this option.

`TCP/IP networking (CONFIG_INET) [Y/n/?]` **Press Enter**
Another very important and vital option under Linux. This option allows us to enable support for TCP/IP networking on the computer. TCP/IP are the protocols used on the Internet and on most local Ethernets. It is highly recommended to say `Y` here even if you are not connected to the Internet.

`IP: multicasting (CONFIG_IP_MULTICAST) [Y/n/?]` **n**
This option allows us to enable a code that addresses several networked computers at once. You need it if you intend to participate in the MBONE, a high bandwidth network on top of the Internet which carries audio and video broadcasts. For most people, it's safe to say `N` here.

`IP: advanced router (CONFIG_IP_ADVANCED_ROUTER) [N/y/?]` **n**
This option allows us to configure our Linux system to run mostly as a router. The answer to this question won't directly affect the kernel: answering `N` will just cause the configuration to skip all the questions about advanced routing. In many cases, we can safety keep the default value of `N` here. Only users that want to run their Linux system primarily as a router will answer to this question by `Y` to be presented a list of advanced routing features to enable or reject. If you want to configure your Linux server as a Gateway server, you need to answer `Y` to this question and all questions related to this option.

`IP: kernel level autoconfiguration (CONFIG_IP_PNP) [N/y/?]` **Press Enter**
This option must be set to `Y` only for diskless machines requiring network access to boot. For most people, it's safe to say `N` here.

`IP: tunneling (CONFIG_NET_IPIP) [N/y/?]` **Press Enter**
This option will enable Tunneling on our system. Tunneling is a means of encapsulating data of one protocol type within another protocol and sending it over a channel that understands the encapsulating protocol. This is an advanced feature and only advanced users who know why they need it must answer `Y` to this question.

`IP: GRE tunnels over IP (CONFIG_NET_IPGRE) [N/y/?]` **Press Enter**
This option will enable another kind of Tunneling feature on our system. This method is known as `GRE` (**G**eneric **R**outing **E**ncapsulation). This is an advanced feature and only advanced users that know why they need it must answer `Y` to this question.

`IP: TCP Explicit Congestion Notification support (CONFIG_INET_ECN) [N/y/?]`
Press Enter
This option enables **E**xplicit **C**ongestion **N**otification on your system. `ECN` allows routers to notify clients about network congestion, resulting in fewer dropped packets and increased network performance. This is a very good feature but, unfortunately, there are many broken firewalls on the Internet, which refuse connections from `ECN`-enabled machines, and it may be a while before these firewalls are fixed. Until then, to access a site behind such a firewall you will have to disable this option by saying `N` here.

```
IP: TCP syncookie support (disabled per default) (CONFIG_SYN_COOKIES) [N/y/?] y
```

This option is very important and every one must answer to this question by `Y` because normal TCP/IP networking is open to an attack known as "SYN flooding". This denial-of-service attack prevents legitimate remote users from being able to connect to your computer during an ongoing attack and requires very little work from the attacker, who can operate from anywhere on the Internet. SYN cookies provide protection against this type of attack. Therefore don't forget to answer `Y` to this question.

```
*
* IP: Netfilter Configuration
```

All questions under the "`IP: Netfilter Configuration`" section of the Kernel configuration are related to packet filter firewall support and features. We recommend you enable everything. Below, we show you the answer for each question without any explanation on the features. If you need to get more information about the features that you don't understand, you can simply type **? [Enter]** at the prompt to get help.

```
*
Connection tracking (required for masq/NAT) (CONFIG_IP_NF_CONNTRACK) [N/y/?]
(NEW) y
FTP protocol support (CONFIG_IP_NF_FTP) [N/y/?] (NEW) y
IRC protocol support (CONFIG_IP_NF_IRC) [N/y/?] (NEW) y
IP tables support (required for filtering/masq/NAT) (CONFIG_IP_NF_IPTABLES)
[N/y/?] (NEW) y
limit match support (CONFIG_IP_NF_MATCH_LIMIT) [N/y/?] (NEW) y
MAC address match support (CONFIG_IP_NF_MATCH_MAC) [N/y/?] (NEW) y
netfilter MARK match support (CONFIG_IP_NF_MATCH_MARK) [N/y/?] (NEW) y
Multiple port match support (CONFIG_IP_NF_MATCH_MULTIPORT) [N/y/?] (NEW) y
TOS match support (CONFIG_IP_NF_MATCH_TOS) [N/y/?] (NEW) y
LENGTH match support (CONFIG_IP_NF_MATCH_LENGTH) [N/y/?] (NEW) y
TTL match support (CONFIG_IP_NF_MATCH_TTL) [N/y/?] (NEW) y
tcpmss match support (CONFIG_IP_NF_MATCH_TCPMSS) [N/y/?] (NEW) y
Connection state match support (CONFIG_IP_NF_MATCH_STATE) [N/y/?] (NEW) y
Packet filtering (CONFIG_IP_NF_FILTER) [N/y/?] (NEW) y
REJECT target support (CONFIG_IP_NF_TARGET_REJECT) [N/y/?] (NEW) y
Full NAT (CONFIG_IP_NF_NAT) [N/y/?] (NEW) y
Packet mangling (CONFIG_IP_NF_MANGLE) [N/y/?] (NEW) y
TOS target support (CONFIG_IP_NF_TARGET_TOS) [N/y/?] (NEW) y
MARK target support (CONFIG_IP_NF_TARGET_MARK) [N/y/?] (NEW) y
LOG target support (CONFIG_IP_NF_TARGET_LOG) [N/y/?] (NEW) y
TCPMSS target support (CONFIG_IP_NF_TARGET_TCPMSS) [N/y/?] (NEW) y
```

WARNING: If you want to enable `IPTables` support into the kernel, the `iptables` program must be installed first or you will receive error messages during kernel compilation. This is because when `IPTables` support is enabled, the kernel will associate some part of the `iptables` program with it configuration. Therefore don't forget to install `IPTables` before configuring kernel with `IPTables` support. Finally the same warning is true for `quota` support into the kernel.

```
*
*
* The IPX protocol (CONFIG_IPX) [N/y/?] Press Enter
```

This option allows us to enable Novell networking protocol support on Linux. You need it if you want to access Novell NetWare file or print servers using Linux or if you want to configure your Linux system to run as a Novell NetWare file or print server. In most cases, this is not required and we can answer `N` to this question.

```
Appletalk protocol support (CONFIG_ATALK) [N/y/?] Press Enter
```
This option allows us to enable AppleTalk protocol support on Linux. You need it if you want to access Apple computers using Linux. In most cases, this is not required and we can answer N to this question.

```
DECnet Support (CONFIG_DECNET) [N/y/?] Press Enter
```
This option allows us to enable DECnet networking protocol support on Linux. The DECnet networking protocol was used in many products made by Digital (now Compaq). In most cases, this is not required and we can answer N to this question.

```
802.1d Ethernet Bridging (CONFIG_BRIDGE) [N/y/?] Press Enter
```
This option will allow your Linux system to act as an Ethernet bridge, which means that the different Ethernet segments it is connected to will appear as one Ethernet to the participants. Several such bridges can work together to create even larger networks of Ethernets using the IEEE 802.1 spanning tree algorithm. In most cases, this is not required and we can answer N to this question.

```
*
* QoS and/or fair queueing
* QoS and/or fair queueing (CONFIG_NET_SCHED) [N/y/?] Press Enter
```
This option allows us to enable QoS (**Q**uality **o**f **S**ervice) support on Linux. When the kernel has several packets to send out over a network device, it has to decide which ones to send first, which ones to delay, and which ones to drop. This is the job of the packet scheduler, and several different algorithms for how to do this "fairly" have been proposed. If we answer N to this question, the standard packet scheduler, which is a FIFO (first come, first served) will be used by default. The standard packet scheduler is enough for most of us and if you are running a router system or are an advanced user who wants to experiment in some new way with TCP/IP networking, then you can say Y to this question and be able to choose from among several alternative algorithms which can then be attached to different network devices. In most cases, we say N to this question.

```
*
* Telephony Support
* Linux telephony support (CONFIG_PHONE) [N/y/?] Press Enter
```
This option allows us to use a regular phone for voice-over-IP applications. This also means that you have to have a telephony card attached to your computer and you know what to do. Most people will simply answer N to this question.

```
*
* ATA/IDE/MFM/RLL support
* ATA/IDE/MFM/RLL support (CONFIG_IDE) [Y/n/?] Press Enter
```
This option allows the kernel to manage low cost mass storage units such as ATA/(E)IDE and ATAPI units. The most common cases are IDE hard drives and ATAPI CD-ROM drives and since we all have one of these devices attached to our computer we can safety say Y to this question. The only time that you can answer to this question by N is when you know that your system is pure SCSI.

```
*
* IDE, ATA and ATAPI Block devices
* Enhanced IDE/MFM/RLL disk/cdrom/tape/floppy support (CONFIG_BLK_DEV_IDE)
[Y/n/?] Press Enter
```
This option allows us to enable the new enhanced driver with IDE/MFM/RLL disk/cdrom/tape/floppy drives. If you have one or more IDE drives, it is required to answer Y to this question.

`Use old disk-only driver on primary interface (CONFIG_BLK_DEV_HD_IDE) [N/y/?]` **Press Enter**
This option allows us to enable the old hard disk driver to control `IDE/MFM/RLL` disk/cdrom/tape/floppy drives. It is highly recommended not to enable it, since we've already used the new enhanced driver option with `IDE/MFM/RLL` disk/cdrom/tape/floppy drives above. This option may be useful for older systems.

`Include IDE/ATA-2 DISK support (CONFIG_BLK_DEV_IDEDISK) [Y/n/?]` **Press Enter**
This option allows us to enable another enhanced support for `MFM/RLL/IDE` hard disks. The only time that you can answer `N` to this question is when you know that your system is pure `SCSI`. Therefore, we'll enable support for this option by saying `Y` to the question.

`Use multi-mode by default (CONFIG_IDEDISK_MULTI_MODE) [Y/n/?]` **n**
This option allows us to fix possible error messages that can appear on `IDE` systems. This error message may look like:

```
hda: set_multmode: status=0x51 { DriveReady SeekComplete Error }
hda: set_multmode: error=0x04 { DriveStatusError }
```

If you get this kind of error message on your system, then you have to say `Y` to this option. We suppose that you have a good `IDE` disk drive and that this error will never appear for you, in this case, we will change the default value of `Y` to become `N`.

`Include IDE/ATAPI CDROM support (CONFIG_BLK_DEV_IDECD) [Y/n/?]` **n**
This option allows us to instruct Linux to identify more than one CD-ROM drive along with other `IDE` devices, as "`hdb`" or "`hdc`", or something similar at boot time. If you have only one CD-ROM drive installed on your system, you can say `N` to this question even if it uses the `ATAPI` protocol and save some KB into the kernel. You have to say `Y` to this option only if you handle more than one CD-ROM drive in your computer. Since most people will usually have only one CD-ROM installed into their computer, we will change the default value of `Y` to become `N`. If you have one CD-ROM and another one CD-ROM for burning disk, then you have answer `Y` to this question.

`Include IDE/ATAPI TAPE support (CONFIG_BLK_DEV_IDETAPE) [N/y/?]` **Press Enter**
This option allows us to enable an `IDE` tape drive using the `ATAPI` protocol. If you have an `IDE` tape drive installed on your system, you must answer `Y` to this question. Most people will say `N` here.

`Include IDE/ATAPI FLOPPY support (CONFIG_BLK_DEV_IDEFLOPPY) [N/y/?]` **Press Enter**
This option allows us to enable an `IDE` floppy drive which uses the `ATAPI` protocol. If you have this kind of floppy drive installed on your system, you must answer this question `Y`. Most people will say `N` here.

`SCSI emulation support (CONFIG_BLK_DEV_IDESCSI) [N/y/?]` **Press Enter**
This option provides `SCSI` host adapter emulation for `IDE ATAPI` devices, and will allow us to use a `SCSI` device driver instead of a native `ATAPI` driver. If you intend to install and use a CD-RW drive on your computer, then you have to say `Y` here. Again, most people will say `N` here.

```
*
* IDE chipset support/bugfixes
* CMD640 chipset bugfix/support (CONFIG_BLK_DEV_CMD640) [Y/n/?] n
```

This option allows us to include code which tries to automatically detect and correct the CMD640 problems under Linux. The CMD-Technologies CMD640 IDE chip is used on many common 486 and Pentium motherboards, usually in combination with a "Neptune" or "SiS" chipset. Unfortunately, it has a number of rather nasty design flaws that can cause severe data corruption under many common conditions. To know if you need to enable this option for your system to correct this bug, edit the /proc/cpuinfo file and see if the parameter "f00f_bug" is set to no or yes. If the "f00f_bug" value is set to no, then you don't need to enable this option and can say N to the question, otherwise you have to say Y here.

```
RZ1000 chipset bugfix/support (CONFIG_BLK_DEV_RZ1000) [Y/n/?] n
```

This option allows us to include code which tries to automatically detect and correct the RZ1000 problems under Linux. The PC-Technologies RZ1000 IDE chip is used on many common 486 and Pentium motherboards, usually along with the "Neptune" chipset. As for the CMD640 bug above, it also has a number of rather nasty design flaws that can cause severe data corruption under many common conditions. To know if you need to enable this option for your system to correct this bug, edit the /proc/cpuinfo file and see if the parameter "coma_bug" is set to no or yes. If the "coma_bug" value is set to no, then you don't need to enable this option and can say N to the question, otherwise you have to say Y here.

```
Generic PCI IDE chipset support (CONFIG_BLK_DEV_IDEPCI) [Y/n/?] Press Enter
```

This option helps the IDE driver to automatically detect and configure all PCI-based IDE interfaces in your system. If you have PCI systems which use IDE drive(s), then say Y to this question. Most of us have PCI systems which use IDE and have answer Y this question.

```
Sharing PCI IDE interrupts support (CONFIG_IDEPCI_SHARE_IRQ) [Y/n/?] Press
Enter
```

This option allows us to enable support for sharing a single IRQ with other cards under ATA/IDE chipsets. In general, everyone has answer Y this question.

```
Generic PCI bus-master DMA support (CONFIG_BLK_DEV_IDEDMA_PCI) [Y/n/?] Press
Enter
```

This option allows us to reduce CPU overhead with IDE drive(s) on PCI system capable of bus-master DMA operation. If you have a modern IDE ATA/33/66/100 hard drive, then it is recommended to answer this question Y.

```
Boot off-board chipsets first support (CONFIG_BLK_DEV_OFFBOARD) [N/y/?] Press
Enter
```

This option allows us to reverse the device scan order to improve the usability of some boot managers such as lilo when booting from a drive on an off-board controller. In many cases, this option is really not required and you can accept the default value of N here.

```
Use PCI DMA by default when available (CONFIG_IDEDMA_PCI_AUTO) [Y/n/?] Press
Enter
```

A very important option to enable on all modern IDE disk drives. This option allows us to use the DMA feature of our disk drive under Linux to improve performance. Most people running a capable DMA drive will answer to this question by Y.

All of the following Kernel options are related to the special onboard chipsets that you may have on your motherboard. Therefore, specific drivers are provided for each of them and you have to choose from the list the one that matches you chipset. If you have an Intel onboard chipset, then you can safety choose the default answer of N to all of the questions, since the kernel supports it naturally.

Other chipset models must be selected from the list. In many cases, if your chipset is not listed, this means that it is automatically supported by the Kernel. Note that two options have their default answer set to Y (`Intel PIIXn chipsets support (CONFIG_BLK_DEV_PIIX) [Y/n/?]` and `PIIXn Tuning support (CONFIG_PIIX_TUNING) [Y/n/?]`). If you have a Pentium II or later processor, you must keep the default value of these two option to Y.

```
AEC62XX chipset support (CONFIG_BLK_DEV_AEC62XX) [N/y/?] Press Enter
ALI M15x3 chipset support (CONFIG_BLK_DEV_ALI15X3) [N/y/?] Press Enter
CMD64X chipset support (CONFIG_BLK_DEV_CMD64X) [N/y/?] Press Enter
CY82C693 chipset support (CONFIG_BLK_DEV_CY82C693) [N/y/?] Press Enter
Cyrix CS5530 MediaGX chipset support (CONFIG_BLK_DEV_CS5530) [N/y/?] Press
Enter
HPT34X chipset support (CONFIG_BLK_DEV_HPT34X) [N/y/?] Press Enter
HPT366 chipset support (CONFIG_BLK_DEV_HPT366) [N/y/?] Press Enter
Intel PIIXn chipsets support (CONFIG_BLK_DEV_PIIX) [Y/n/?] Press Enter
PIIXn Tuning support (CONFIG_PIIX_TUNING) [Y/n/?] Press Enter
NS87415 chipset support (EXPERIMENTAL) (CONFIG_BLK_DEV_NS87415) [N/y/?] Press
Enter
PROMISE PDC202{46|62|65|67|68} support (CONFIG_BLK_DEV_PDC202XX) [N/y/?] Press
Enter
ServerWorks OSB4/CSB5 chipsets support (CONFIG_BLK_DEV_SVWKS) [N/y/?] Press
Enter
SiS5513 chipset support (CONFIG_BLK_DEV_SIS5513) [N/y/?] Press Enter
SLC90E66 chipset support (CONFIG_BLK_DEV_SLC90E66) [N/y/?] Press Enter
Tekram TRM290 chipset support (EXPERIMENTAL) (CONFIG_BLK_DEV_TRM290) [N/y/?]
Press Enter
VIA82CXXX chipset support (CONFIG_BLK_DEV_VIA82CXXX) [N/y/?] Press Enter
Other IDE chipset support (CONFIG_IDE_CHIPSETS) [N/y/?] Press Enter
IGNORE word93 Validation BITS (CONFIG_IDEDMA_IVB) [N/y/?] Press Enter

*
* SCSI support
* SCSI support (CONFIG_SCSI) [Y/n/?] Press Enter
```

This option allows us to enable SCSI hard disks, SCSI tape drives, SCSI CD-ROM's or any other SCSI devices under Linux. If you have a SCSI like system, you need to answer Y to this question. If you don't have any SCSI devices on your system, you can safety answer N to the question. For users that have a SCSI system, it is very important for you to know the name of your SCSI host adapter (the card inside your computer that "speaks" the SCSI protocol, also called SCSI controller), because you will be asked for it if you enable this option. Once again, if you don't have a SCSI system, simply answer N to this question and skip this section of the Linux Kernel configuration.

```
*
* SCSI support type (disk, tape, CD-ROM)
* SCSI disk support (CONFIG_BLK_DEV_SD) [Y/n/?] Press Enter
```

This option allows us to enable support for a SCSI hard disk under Linux. If you have enabled the SCSI support feature above because you have a SCSI hard drive on your system, then it's here that you have to specify it by answering Y to the question.

```
Maximum number of SCSI disks that can be loaded as modules
(CONFIG_SD_EXTRA_DEVS) [40] Press Enter
```
This option allows us to control the amount of additional space allocated in tables for drivers that are loaded as modules after the kernel is booted. In the event that the SCSI core itself was loaded as a module, this value is the number of additional disks that can be loaded after the first host driver is loaded. Since we're compiling a Monolithic Kernel where no modules are available, this option doesn't concern us and we can safety press the [Enter] key to accept the default value.

```
SCSI tape support (CONFIG_CHR_DEV_ST) [N/y/?] Press Enter
```
This option allows us to enable support for a SCSI tape drive under Linux. If you have enabled the SCSI support feature above because you have a SCSI tape drive on your system, then it's here that you have to specify it by answering Y to the question. For most SCSI users, the answer is N (no, we don't have a SCSI tape drive on this computer).

```
SCSI OnStream SC-x0 tape support (CONFIG_CHR_DEV_OSST) [N/y/?] Press Enter
```
This option allows us to enable support for the OnStream SC-x0 SCSI tape drives under Linux. If you have this kind of SCSI tape drive installed on your computer, then you have to answer to this question by Y. Most SCSI users will simply say N to this question.

```
SCSI CD-ROM support (CONFIG_BLK_DEV_SR) [N/y/?] Press Enter
```
This option allows us to enable support for a SCSI CD-ROM under Linux. If you have enabled the SCSI support feature above because you have a SCSI CD-ROM on your system, then it's here that you have to specify it by answering Y to the question. For most SCSI users, the answer is N (no, we don't have a SCSI CD-ROM on this computer).

```
SCSI generic support (CONFIG_CHR_DEV_SG) [N/y/?] Press Enter
```
This option allows us to enable support to use SCSI scanners, synthesizers or CD-writers or just about anything having "SCSI" in its name other than hard disks, CD-ROMs or tapes. If you have one of these SCSI items installed on your computer, then you have to say Y here as well as for the "SCSI disk support" option above to enable the driver.

> **VERY IMPORTANT NOTE:** For users having IDE CD-writers, you have to say Y to this question too, even if your CD-writers are not SCSI CD-writers. Most SCSI users will simply say N to his question.

```
*
* Some SCSI devices (e.g. CD jukebox) support multiple LUNs
* Enable extra checks in new queueing code (CONFIG_SCSI_DEBUG_QUEUES) [Y/n/?]
Press Enter
```
This option turns on a lot of additional consistency checking for the new queuing code on SCSI devices. It is a good idea to enable it by saying Y to the question.

```
Probe all LUNs on each SCSI device (CONFIG_SCSI_MULTI_LUN) [Y/n/?] n
```
This option force the SCSI driver to probe for multiple LUN's (**L**ogical **U**nit **N**umber) on your system and will certainly affect the performance of the system. Most SCSI users will simply disable this option by saying N to this question to improve performance. If you enable this option, then we assume that you know what you're doing.

```
Verbose SCSI error reporting (kernel size +=12K) (CONFIG_SCSI_CONSTANTS)
[Y/n/?] n
```

This option allows any error messages regarding your SCSI hardware to be more understandable, this enlarges your kernel by about 12 KB. If performance is important to you, we highly recommend you to disable this option by answering the question N.

```
SCSI logging facility (CONFIG_SCSI_LOGGING) [N/y/?] Press Enter
```

This option allows us to turns on a logging facility that can be used to debug a number of SCSI related problems. Again, if performance is important to you, we highly recommend you to disable this option by keeping the default value of N here.

```
*
* SCSI low-level drivers
```

Below you will be presented a list of available SCSI controllers to choose from, simply select the SCSI controller that is installed on your system and disable all the others. As an example, we will pretend that we have an Adaptec AIC7080 controller and will enable it further down. We chose an Adaptec AIC7080 model for our example; don't forget to change our choice if you have another kind of SCSI controller installed on your system.

```
*
3ware Hardware ATA-RAID support (CONFIG_BLK_DEV_3W_XXXX_RAID) [N/y/?] Press
Enter
7000FASST SCSI support (CONFIG_SCSI_7000FASST) [N/y/?] Press Enter
ACARD SCSI support (CONFIG_SCSI_ACARD) [N/y/?] Press Enter
Adaptec AHA152X/2825 support (CONFIG_SCSI_AHA152X) [N/y/?] Press Enter
Adaptec AHA1542 support (CONFIG_SCSI_AHA1542) [N/y/?] Press Enter
Adaptec AHA1740 support (CONFIG_SCSI_AHA1740) [N/y/?] Press Enter
Adaptec AIC7xxx support (CONFIG_SCSI_AIC7XXX) [N/y/?] y
Maximum number of TCQ commands per device (CONFIG_AIC7XXX_CMDS_PER_DEVICE)
[253] (NEW) Press Enter
Initial bus reset delay in milli-seconds (CONFIG_AIC7XXX_RESET_DELAY_MS)
[15000] (NEW) Press Enter
Build Adapter Firmware with Kernel Build (CONFIG_AIC7XXX_BUILD_FIRMWARE)
[N/y/?] (NEW) Press Enter
Adaptec I2O RAID support (CONFIG_SCSI_DPT_I2O) [N/y/?] Press Enter
AdvanSys SCSI support (CONFIG_SCSI_ADVANSYS) [N/y/?] Press Enter
Always IN2000 SCSI support (CONFIG_SCSI_IN2000) [N/y/?] Press Enter
AM53/79C974 PCI SCSI support (CONFIG_SCSI_AM53C974) [N/y/?] Press Enter
AMI MegaRAID support (CONFIG_SCSI_MEGARAID) [N/y/?] Press Enter
BusLogic SCSI support (CONFIG_SCSI_BUSLOGIC) [N/y/?] Press Enter
Compaq Fibre Channel 64-bit/66Mhz HBA support (CONFIG_SCSI_CPQFCTS) [N/y/?]
Press Enter
DMX3191D SCSI support (CONFIG_SCSI_DMX3191D) [N/y/?] Press Enter
DTC3180/3280 SCSI support (CONFIG_SCSI_DTC3280) [N/y/?] Press Enter
EATA ISA/EISA/PCI (DPT and generic EATA/DMA-compliant boards) support
(CONFIG_SCSI_EATA) [N/y/?] Press Enter
EATA-DMA [Obsolete] (DPT, NEC, AT&T, SNI, AST, Olivetti, Alphatronix) support
(CONFIG_SCSI_EATA_DMA) [N/y/?] Press Enter
EATA-PIO (old DPT PM2001, PM2012A) support (CONFIG_SCSI_EATA_PIO) [N/y/?] Press
Enter
Future Domain 16xx SCSI/AHA-2920A support (CONFIG_SCSI_FUTURE_DOMAIN) [N/y/?]
Press Enter
Intel/ICP (former GDT SCSI Disk Array) RAID Controller support
(CONFIG_SCSI_GDTH) [N/y/?] Press Enter
Generic NCR5380/53c400 SCSI support (CONFIG_SCSI_GENERIC_NCR5380) [N/y/?] Press
Enter
IBM ServeRAID support (CONFIG_SCSI_IPS) [N/y/?] Press Enter
Initio 9100U(W) support (CONFIG_SCSI_INITIO) [N/y/?] Press Enter
Initio INI-A100U2W support (CONFIG_SCSI_INIA100) [N/y/?] Press Enter
```

```
NCR53c406a SCSI support (CONFIG_SCSI_NCR53C406A) [N/y/?] Press Enter
NCR53c7,8xx SCSI support (CONFIG_SCSI_NCR53C7xx) [N/y/?] Press Enter
SYM53C8XX Version 2 SCSI support (CONFIG_SCSI_SYM53C8XX_2) [N/y/?] Press Enter
NCR53C8XX SCSI support (CONFIG_SCSI_NCR53C8XX) [N/y/?] Press Enter
SYM53C8XX SCSI support (CONFIG_SCSI_SYM53C8XX) [Y/n/?] n
PAS16 SCSI support (CONFIG_SCSI_PAS16) [N/y/?] Press Enter
PCI2000 support (CONFIG_SCSI_PCI2000) [N/y/?] Press Enter
PCI2220i support (CONFIG_SCSI_PCI2220I) [N/y/?] Press Enter
PSI240i support (CONFIG_SCSI_PSI240I) [N/y/?] Press Enter
Qlogic FAS SCSI support (CONFIG_SCSI_QLOGIC_FAS) [N/y/?] Press Enter
Qlogic ISP SCSI support (CONFIG_SCSI_QLOGIC_ISP) [N/y/?] Press Enter
Qlogic ISP FC SCSI support (CONFIG_SCSI_QLOGIC_FC) [N/y/?] Press Enter
Qlogic QLA 1280 SCSI support (CONFIG_SCSI_QLOGIC_1280) [N/y/?] Press Enter
Seagate ST-02 and Future Domain TMC-8xx SCSI support (CONFIG_SCSI_SEAGATE)
[N/y/?] Press Enter
Simple 53c710 SCSI support (Compaq, NCR machines) (CONFIG_SCSI_SIM710) [N/y/?]
Press Enter
Symbios 53c416 SCSI support (CONFIG_SCSI_SYM53C416) [N/y/?] Press Enter
Tekram DC390(T) and Am53/79C974 SCSI support (CONFIG_SCSI_DC390T) [N/y/?] Press
Enter
Trantor T128/T128F/T228 SCSI support (CONFIG_SCSI_T128) [N/y/?] Press Enter
UltraStor 14F/34F support (CONFIG_SCSI_U14_34F) [N/y/?] Press Enter
UltraStor SCSI support (CONFIG_SCSI_ULTRASTOR) [N/y/?] Press Enter

*
* Fusion MPT device support
* Fusion MPT (base + ScsiHost) drivers (CONFIG_FUSION) [N/y/?] Press Enter

*
* I2O device support
* I2O support (CONFIG_I2O) [N/y/?] Press Enter
```

This option allows us to enable support for Intelligent Input/Output (I2O) architecture. In order for this to work, you need to have an I2O interface adapter card in your computer. If you have this kind of I2O interface adapter card installed on your system, then you can say Y to the question and you will get a choice of interface adapter drivers and OSM's. Most users simply say N here.

```
*
* Network device support
* Network device support (CONFIG_NETDEVICES) [Y/n/?] Press Enter
```

This option is one of the most important and allows us to enable support and feature for network cards under Linux. Therefore, we have to answer `Y` to this question.

```
*
* ARCnet devices
* ARCnet support (CONFIG_ARCNET) [N/y/?] Press Enter
```

This option allows us to enable ARCnet chipset support under Linux. If you have a network card of this type installed on your system, then say `Y` here, otherwise and for most users, you have to keep the default value of `N` here.

```
Dummy net driver support (CONFIG_DUMMY) [Y/n/?] n
```

This option is only useful for `PPP` dial up modem users on Linux. If you don't use your system to make `PPP` connections, then you can safety answer `N` to this question. Only users who have a modem and want to use it to establish a connection via `PPP` or `SLIP` need to say `Y` here. Since, in our example, we use a network card to make a network connection, we will answer the question `N`.

```
Bonding driver support (CONFIG_BONDING) [N/y/?] Press Enter
```
This option allows us to 'bond' multiple Ethernet Channels together. This is called 'Etherchannel' by Cisco, 'Trunking' by Sun, and 'Bonding' in Linux. It is a technique to merge Ethernet segments together for doubling the speed of a connection. In most cases, we can safety choose the default choice of N here. You must have two Ethernet cards installed on your system and on the remote computer where you want to use this technique to be able to use it. Also, this is an advanced feature and only experienced Linux users may need it. Therefore, we will answer to the question by N.

```
EQL (serial line load balancing) support (CONFIG_EQUALIZER) [N/y/?] Press Enter
```
This option allows us to enable the same feature as the previous option, but this time for two modems and two telephone lines. Therefore, we will simply say N to this question.

```
Universal TUN/TAP device driver support (CONFIG_TUN) [N/y/?] Press Enter
```
This option allows us to enable TUN/TAP support under Linux. TUN/TAP provides packet reception and transmission for user space programs. It can be viewed as a simple Point-to-Point or Ethernet device, which instead of receiving packets from a physical media, receives them from user space program and instead of sending packets via physical media, writes them to the user space program. It is rare that we need this feature, if you need it then simply say Y here. Most users will say N here.

```
*
* Ethernet (10 or 100Mbit)
* Ethernet (10 or 100Mbit) (CONFIG_NET_ETHERNET) [Y/n/?] Press Enter
```
This option is very important and allows us to enable support for Ethernet Network Interface Cards (NIC's) under Linux. Now, everyone has a NIC in their computer and if you want to be able to use your network card, then you have to say Y here. Note that the answer to this question won't directly affect the kernel: saying N will just cause the configuration to skip all the questions about Ethernet network cards. We must say Y here to be able to select the network card that we have in our computer from the list of supported network cards.

It is very important to know the name of the network card(s) installed in your system because you will be asked for it. As an example we will pretend that we have an "EtherExpressPro/100" network card in our computer and we will enable support for it. This is an example and don't forget to change our default choice if you have another kind of network card installed in your system. In general, we say Y for the network card that we have and N for all other network cards.

```
Sun Happy Meal 10/100baseT support (CONFIG_HAPPYMEAL) [N/y/?] Press Enter
Sun GEM support (CONFIG_SUNGEM) [N/y/?] Press Enter
3COM cards (CONFIG_NET_VENDOR_3COM) [N/y/?] Press Enter
AMD LANCE and PCnet (AT1500 and NE2100) support (CONFIG_LANCE) [N/y/?] Press
Enter
Western Digital/SMC cards (CONFIG_NET_VENDOR_SMC) [N/y/?] Press Enter
Racal-Interlan (Micom) NI cards (CONFIG_NET_VENDOR_RACAL) [N/y/?] Press Enter
DEPCA, DE10x, DE200, DE201, DE202, DE422 support (CONFIG_DEPCA) [N/y/?] Press
Enter
HP 10/100VG PCLAN (ISA, EISA, PCI) support (CONFIG_HP100) [N/y/?] Press Enter
Other ISA cards (CONFIG_NET_ISA) [N/y/?] Press Enter
EISA, VLB, PCI and on board controllers (CONFIG_NET_PCI) [Y/n/?] Press Enter
AMD PCnet32 PCI support (CONFIG_PCNET32) [N/y/?] Press Enter
Apricot Xen-II on board Ethernet (CONFIG_APRICOT) [N/y/?] Press Enter
CS89x0 support (CONFIG_CS89x0) [N/y/?] Press Enter
DECchip Tulip (dc21x4x) PCI support (CONFIG_TULIP) [N/y/?] Press Enter
Generic DECchip & DIGITAL EtherWORKS PCI/EISA (CONFIG_DE4X5) [N/y/?] Press
Enter
Digi Intl. RightSwitch SE-X support (CONFIG_DGRS) [N/y/?] Press Enter
Davicom DM910x/DM980x support (CONFIG_DM9102) [N/y/?] Press Enter
```

```
EtherExpressPro/100 support (CONFIG_EEPRO100) [Y/n/?] Press Enter
Myson MTD-8xx PCI Ethernet support (CONFIG_FEALNX) [N/y/?] Press Enter
National Semiconductor DP8381x series PCI Ethernet support (CONFIG_NATSEMI)
[N/y/?] Press Enter
PCI NE2000 and clones support (see help) (CONFIG_NE2K_PCI) [N/y/?] Press Enter
RealTek RTL-8139 PCI Fast Ethernet Adapter support (CONFIG_8139TOO) [N/y/?]
Press Enter
SiS 900/7016 PCI Fast Ethernet Adapter support (CONFIG_SIS900) [N/y/?] Press
Enter
SMC EtherPower II (CONFIG_EPIC100) [N/y/?] Press Enter
Sundance Alta support (CONFIG_SUNDANCE) [N/y/?] Press Enter
TI ThunderLAN support (CONFIG_TLAN) [N/y/?] Press Enter
VIA Rhine support (CONFIG_VIA_RHINE) [N/y/?] Press Enter
Winbond W89c840 Ethernet support (CONFIG_WINBOND_840) [N/y/?] Press Enter
Pocket and portable adapters (CONFIG_NET_POCKET) [N/y/?] Press Enter
*
* Ethernet (1000 Mbit)
* Alteon AceNIC/3Com 3C985/NetGear GA620 Gigabit support (CONFIG_ACENIC)
[N/y/?] Press Enter
D-Link DL2000-based Gigabit Ethernet support (CONFIG_DL2K) [N/y/?] Press Enter
National Semiconduct DP83820 support (CONFIG_NS83820) [N/y/?] Press Enter
Packet Engines Hamachi GNIC-II support (CONFIG_HAMACHI) [N/y/?] Press Enter
SysKonnect SK-98xx support (CONFIG_SK98LIN) [N/y/?] Press Enter

FDDI driver support (CONFIG_FDDI) [N/y/?] Press Enter
```

This option allows us to enable FDDI card support under Linux. FDDI (**F**iber **D**istributed **D**ata **I**nterface) is a high speed local area network design to runs over copper or fiber. If you are connected to such a network and want a driver for the FDDI card in your computer, say Y here. Most users will simply say N here.

```
PPP (point-to-point protocol) support (CONFIG_PPP) [N/y/?] Press Enter
```

This option allows us to enable PPP support under Linux. PPP (**P**oint to **P**oint **P**rotocol) is the protocol used by modern modems to establish a remote connection with your ISP. If you have a modem card installed on your system to make a remote connection with your ISP, then you need to answer Y to this question. If you don't use PPP to connect on the Internet, then you can safety say N here. In our example, we assume that you use another method, like a network interface, to connect to the Internet and say N here.

```
SLIP (serial line) support (CONFIG_SLIP) [N/y/?] Press Enter
```

This option allows us to enable SLIP or CSLIP support under Linux. These protocols are really old now and have been replaced by the PPP protocol (see the above option). If for any reason you still use them, then say Y here. Most users will answer this question N.

```
*
* Wireless LAN (non-hamradio)
* Wireless LAN (non-hamradio) (CONFIG_NET_RADIO) [N/y/?] Press Enter
```

This option allows us to enable support for wireless LANs and everything having to do with radio, but not with amateur radio or FM broadcasting. If you need such support on your system, then say Y here, also if you need Wireless Extensions with wireless PCMCIA (PC-) cards on Linux, you need to say Y here too. Most users will simply say N here.

```
*
* Token Ring devices
* Token Ring driver support (CONFIG_TR) [N/y/?] Press Enter
```

This option allows us to enable support for Token Ring under Linux. Token Ring is IBM's way of communication on a local network; the rest of the world uses Ethernet. If you need Token Ring support on your computer, then say Y here. Most people will select the default choice of N here.

```
Fibre Channel driver support (CONFIG_NET_FC) [N/y/?] Press Enter
```
This option allows us to enable Fibre Channel support under Linux. Fibre Channel is a high speed serial protocol mainly used to connect large storage devices to the computer. If you have a Fibre channel adapter card in your computer, then you can say Y here. Most users will simply say N here.

```
*
* Wan interfaces
* Wan interfaces support (CONFIG_WAN) [N/y/?] Press Enter
```
This option allows us to enable support for Wan interfaces under Linux. **W**ide **A**rea **N**etworks (WANs), such as X.25, frame relay and leased lines, are used to interconnect **L**ocal **A**rea **N**etworks (LANs) over vast distances. If you have these kinds of cards installed on your system, then you can answer Y to the question. Most users will say N here.

```
*
* Amateur Radio support
* Amateur Radio support (CONFIG_HAMRADIO) [N/y/?] Press Enter
```
This option allows us to enable Amateur Radio support under Linux. If you want to connect your Linux box to an amateur radio, answer Y here. Note that the answer to this question won't directly affect the kernel: saying N will just cause the configuration to skip all the questions about amateur radio. Most people will say N here.

```
*
* IrDA (infrared) support
* IrDA subsystem support (CONFIG_IRDA) [N/y/?] Press Enter
```
This option allows us to enable support for the IrDA (TM) protocols under Linux. IrDA (**I**nfra**r**ed **D**ata **A**ssociations) is a support for wireless infrared communication. Laptops or computers that use infrared and PDA's users will say Y here. Most users will say N here.

```
*
* ISDN subsystem
* ISDN support (CONFIG_ISDN) [N/y/?] Press Enter
```
This option allows us to enable ISDN support under Linux. ISDN (**I**ntegrated **S**ervices **D**igital **N**etworks) is a special type of fully digital telephone service; it's mostly used to connect to your Internet service provider (with SLIP or PPP). If you have this type of card installed on your computer (popular in Europe), then you have to say Y here otherwise, you will certainly keep the default value of N here.

```
*
* Old CD-ROM drivers (not SCSI, not IDE)
* Support non-SCSI/IDE/ATAPI CDROM drives (CONFIG_CD_NO_IDESCSI) [N/y/?] Press
Enter
```
If you have a CD-ROM drive that is neither SCSI nor IDE/ATAPI, say Y here, otherwise N. Note that the answer to this question doesn't directly affect the kernel: saying N will just cause the configuration to skip all the questions about these CD-ROM drives. Most users will say N here.

```
*
* Input core support
* Input core support (CONFIG_INPUT) [N/y/?] Press Enter
```
This option allows us to enable any of the USB HID (**H**uman **I**nterface **D**evice) options in the USB support section which require Input core support. If you intended to use USB on Linux, say Y here otherwise, you can safety say N here. It is rare that we have to use USB on server systems.

```
*
* Character devices
* Virtual terminal (CONFIG_VT) [Y/n/?] Press Enter
```
This option is very important and allows us to enable support for terminal devices with display and keyboard devices. On Linux, you need at least one virtual terminal device in order to make use of your keyboard and monitor. Therefore, we have to say `Y` here.

```
Support for console on virtual terminal (CONFIG_VT_CONSOLE) [Y/n/?] Press Enter
```
This option allows us to enable support for consoles on virtual terminals. Most users will simply keep the default value of `Y` here.

```
Standard/generic (8250/16550 and compatible UARTs) serial support
(CONFIG_SERIAL) [Y/n/?] n
```
This option allows us to use serial mice, modems and similar devices that connect to the standard serial ports under Linux. If you use your Linux system as a workstation with graphical interface installed, then you must answer `Y` to the question. If you use your Linux system for dedicated Ethernet WWW/FTP servers, then you can say `N` here. In our example, we assume that your Linux system is dedicated and configured to run as a server only and answer `N` to the question.

```
Non-standard serial port support (CONFIG_SERIAL_NONSTANDARD) [N/y/?] Press
Enter
```
This option allows us to enable support for any non-standard serial boards under Linux. These are usually used for systems that need many serial ports, because they serve many terminals or dial-in connections. If this is not true in your case, then you can safety say `N` here. Most people will simply say `N` here.

```
Unix98 PTY support (CONFIG_UNIX98_PTYS) [Y/n/?] Press Enter
```
This option is important in modern Linux system and allows us to enable pseudo terminal (`PTY`) support under Linux. Everyone must say `Y` here. Information about pseudo terminal (`PTY`) can be found in the Kernel documentation.

```
Maximum number of Unix98 PTYs in use (0-2048) (CONFIG_UNIX98_PTY_COUNT) [256]
128
```
This option allows us to set the maximum number of Unix98 PTY's that can be used at any one time. It is important to note that each additional set of 256 PTY's occupy approximately 8 KB of kernel memory on 32-bit architectures and for security as well as performance reasons, we must keep the number as low as possible. Therefore, we will change the default value of 256 to 128.

```
*
* I2C support
* I2C support (CONFIG_I2C) [N/y/?] Press Enter
```
This option allows us to enable `I2C` and `SMBus` support under Linux. `IC2` is a slow serial bus protocol used in many micro controller applications and developed by Philips. If you need this feature, then say `Y` to the question otherwise say `N` here. Most people will say `N` here.

```
Bus Mouse Support (CONFIG_BUSMOUSE) [N/y/?] Press Enter
```
This option allows us to enable bus mouse support under Linux. All modern computer now use a serial mouse. If you still continue to use a bus mouse on your system, then you have to say `Y` here. Laptop users also need to say `Y` here. Most users will simply say `N` here.

Mouse Support (not serial and bus mice) (CONFIG_MOUSE) [Y/n/?] **Press Enter**
This option allows us to enable support for no serial or bus mice support under Linux. This is for machines with a mouse, which is neither a serial, nor a bus mouse. Examples are PS/2 mice. If you have a PS/2 mouse, then say Y here, otherwise say N here. Laptop and workstation users also need to say Y here. If you use your Linux system for dedicated Ethernet WWW/FTP servers, then you can say N here and save some space in your Kernel code.

PS/2 mouse (aka "auxiliary device") support (CONFIG_PSMOUSE) [Y/n/?] **Press Enter**
This option enables support for a PS/2 mouse on your system. The PS/2 mouse connects to a special mouse port that looks much like the keyboard port (small circular connector with 6 pins). If you have this kind of mouse (most modern computers and laptops have and use it), then say Y here otherwise say N.

C&T 82C710 mouse port support (as on TI Travelmate) (CONFIG_82C710_MOUSE) [N/y/?] **Press Enter**
This option allows us to enable support for a certain kind of PS/2 mouse used on the TI Travelmate. If you have this kind of mouse installed on your system, then say Y here. Most users will say N to this question.

PC110 digitizer pad support (CONFIG_PC110_PAD) [N/y/?] **Press Enter**
This drives the digitizer pad on the IBM PC110 palmtop. It can turn the digitizer pad into PS/2 mouse emulation with tap gestures or into an absolute pad. Most users will answer this question N.

QIC-02 tape support (CONFIG_QIC02_TAPE) [N/y/?] **Press Enter**
This option allows us to enable QIC-02 tape support under Linux. QIC-02 is a non-SCSI tape drive and if you use it, then says Y to the question. Most users will say N here.

*
*** Watchdog Cards**
* Watchdog Timer Support (CONFIG_WATCHDOG) [N/y/?] **Press Enter**
This option enables Watchdog Timer support under Linux. For details about watchdog, please read Documentation/watchdog.txt in the kernel source. Most users will say N here.

Intel i8x0 Random Number Generator support (CONFIG_INTEL_RNG) [N/y/?] **Press Enter**
This option allows us to enable support for a driver that provides kernel-side support for the Random Number Generator hardware found on Intel i8xx-based motherboards. If you have this generator hardware installed on your computer, then you can say Y to the question. Most people will say N here.

/dev/nvram support (CONFIG_NVRAM) [N/y/?] **Press Enter**
This option allows us to get read and write access to the 50 bytes of non-volatile memory in the **r**eal **t**ime **c**lock (**RTC**). This memory is conventionally called "CMOS RAM" on PCs and may be used to view settings there, or to change them (with some utility). Most users will say N to this question.

Enhanced Real Time Clock Support (CONFIG_RTC) [N/y/?] **Press Enter**
This option allows us to get access to the real time clock (or hardware clock) built into the computer. If you run Linux on a multiprocessor machine and said Y to "Symmetric Multi Processing" above, you should say Y here to read and set the RTC in an SMP compatible fashion.

Double Talk PC internal speech card support (CONFIG_DTLK) [N/y/?] **Press Enter**
This option allows us to enable support for the DoubleTalk PC under Linux. If you have a speech card installed on your computer, then answer to this question by Y. Most users will say **N** here.

Siemens R3964 line discipline (CONFIG_R3964) [N/y/?] **Press Enter**
This option allows synchronous communication with devices using the Siemens R3964 packet protocol. Unless you are dealing with special hardware like PLC's, you are unlikely to need this. Most users will simply say N here.

Applicom intelligent fieldbus card support (CONFIG_APPLICOM) [N/y/?] **Press Enter**
This option provides the kernel-side support for the intelligent fieldbus cards made by Applicom International. Unless you are dealing with such a card, you are unlikely to need this. Most users will simply say N here.

*
*** Ftape, the floppy tape device driver**
* Ftape (QIC-80/Travan) support (CONFIG_FTAPE) [N/y/?] **Press Enter**
This option allows us to enable support for some well know tape drives under Linux. If you enable this option, then you'll be asked to select your make and model from the available list. If you don't use tape drive on your computer, then you can say N to this option.

/dev/agpgart (AGP Support) (CONFIG_AGP) [Y/n/?] **n**
This option is only pertinent if you use XFree86 on your computer and want to use the GLX or DRI features for better performance. Enable this option only if you use graphical interface on your computer. If you system is a server, then you really don't need this option. In our example, we are configuring the kernel for a server purpose, therefore, we will say N here.

Direct Rendering Manager (XFree86 4.1.0 and higher DRI support) (CONFIG_DRM) [Y/n/?] **n**
This option is directly related to the use of XFree86 and graphical interface on your computer. It allows us to enable Kernel-level support for the **D**irect **R**endering **I**nfrastructure (DRI) for better performance of the system. If your computer doesn't have a graphical interface installed, and is configured as a server, then you don't need to enable this option. If you say Y here because you have and use a graphical interface, then you need to select the module that's right for your graphics card from the available list. In our example, we are configuring the kernel for a server purpose, therefore, we will say N here.

ACP Modem (Mwave) support (CONFIG_MWAVE) [N/y/?] **Press Enter**
This option allows us to enable ACP modem (Mwave) support under Linux. ACP is a WinModem composed of a kernel driver and a user level application. Together these components support direct attachment to **p**ublic **s**witched **t**elephone **n**etworks (PSTNs) and support selected countries. Most user will simply say N here.

*
*** Multimedia devices**
* Video For Linux (CONFIG_VIDEO_DEV) [N/y/?] **Press Enter**
This option allows us to enable support for audio/video capture and overlay devices and FM radio cards under Linux. If you use graphical interface on your system, you need to say Y here. If your system is configured as a server and you don't have graphical interface installed on it, then you can safety say N here. In our example, we are configuring the kernel for a server purpose, therefore, we will say N here.

```
*
* File systems
* Quota support (CONFIG_QUOTA) [N/y/?] y
```
This option is important to allow us to set per user/group limits for disk usage (also called disk quotas) under Linux. It is sometimes required on server environments where such a purpose is required. If you use Linux as a workstation with graphical interface, it is not useful to enable it. Only enable this option on server environment when required.

```
Kernel automounter support (CONFIG_AUTOFS_FS) [N/y/?] Press Enter
```
This option allows us to automatically mount remote file systems on demand. A newer version of the automounter with more features is now available; therefore we really don't need to enable this option. Say N here.

```
Kernel automounter version 4 support (also supports v3) (CONFIG_AUTOFS4_FS)
[Y/n/?] n
```
This option allows us to enable support for the new automounter tool. If you are not a part of a fairly large, distributed network or don't have a laptop which needs to dynamically reconfigure to the local network, you probably do not need an automounter, and can say N here.

```
Ext3 journalling file system support (EXPERIMENTAL) (CONFIG_EXT3_FS) [N/y/?] y
```
This option is very important and allows us to enable support for the new EXT3 journaling file system support under Linux. EXT3 is the journaling version of the Second extended file system (often called ext3), the de facto standard Linux file system (method to organize files on a storage device) for hard disks. Most users will say Y here to be able to get advantage of the new EXT3 journaling file system.

```
JBD (ext3) debugging support (CONFIG_JBD_DEBUG) [N/y/?] y
```
This option allows us to enable debugging output while the system is running EXT3 under Linux, in order to help track down any problems you are having. It is a good idea to enable it, therefore answer Y to the question.

```
DOS FAT fs support (CONFIG_FAT_FS) [N/y/?] Press Enter
```
If you want to use one of the FAT-based file systems (the MS-DOS, VFAT (Windows 95) and UMSDOS (used to run Linux on top of an ordinary DOS partition) file systems, then you must say Y here. In many case, we really don't need this kind of feature and can safety say N here. Recommended choice is N and especially for a linux server system.

```
Compressed ROM file system support (CONFIG_CRAMFS) [N/y/?] Press Enter
```
This option allows us to enable CramFs support under Linux. CramFs (**C**ompressed **ROM F**ile **S**ystem) is designed to be a simple, small, and compressed file system for ROM based embedded systems. Most users will simply say N here.

```
Virtual memory file system support (former shm fs) (CONFIG_TMPFS) [Y/n/?] Press
Enter
```
This option allows us to enable support for Tmpfs under Linux. Tmpfs is a file system which keeps all files in virtual memory. It is a good idea to enable it on your computer.

```
Simple RAM-based file system support (CONFIG_RAMFS) [N/y/?] Press Enter
```
This option allows us to enable support for Ramfs under Linux. Ramfs is a file system which keeps all files in RAM. It allows read and write access. Most users will say N here.

```
ISO 9660 CDROM file system support (CONFIG_ISO9660_FS) [Y/n/?] Press Enter
```
This option is important and allows Linux to access and read your CD-ROM. Since everyone has and uses a CD-ROM, it is vital to say Y to this question.

Microsoft Joliet CDROM extensions (CONFIG_JOLIET) [N/y/?] **Press Enter**
This option allows us to be able to read Joliet CD-ROM's under Linux. Joliet is a Microsoft extension for the ISO 9660 CD-ROM file system, which allows for long filenames in unicode format. If you think that you'll need to access some files written in this format under Linux with your CD-ROM, then you need to say Y here. For workstations, you may need to say Y here but for Linux servers, you have to say N here.

Transparent decompression extension (CONFIG_ZISOFS) [N/y/?] **Press Enter**
This is a Linux-specific extension to RockRidge which lets you store data in compressed form on a CD-ROM and have it transparently decompressed when the CD-ROM is accessed. Again, for workstations you may need it but for servers, you don't need it. Say N.

Minix fs support (CONFIG_MINIX_FS) [N/y/?] **Press Enter**
This option allows us to enable Minix fs support with Linux. The minix file system (method to organize files on a hard disk partition or a floppy disk) was the original file system for Linux, but has been superseded by the second extended file system (ext2fs). Simply say N here.

FreeVxFS file system support (VERITAS VxFS(TM) compatible) (CONFIG_VXFS_FS) [N/y/?] **Press Enter**
FreeVxFS is a file system driver that support the VERITAS VxFS(TM) file system format. VERITAS VxFS(TM) is the standard file system of SCO UnixWare (and possibly others) and optionally available for Sunsoft Solaris, HP-UX and many other operating systems. If you want to support such file system on your computer, then say Y here otherwise say N.

NTFS file system support (read only) (CONFIG_NTFS_FS) [N/y/?] **Press Enter**
NTFS is the file system of Microsoft Windows NT. Say Y if you want to get read access to files on NTFS partitions of your hard drives otherwise say N here. Most users will say N.

OS/2 HPFS file system support (CONFIG_HPFS_FS) [N/y/?] **Press Enter**
OS/2 is IBM's operating system for PC's, the same as Warp, and HPFS is the file system used for organizing files on OS/2 hard disk partitions. Say Y if you want to be able to read files from and write files to an OS/2 HPFS partition on your hard drive. Most users will say N here.

/proc file system support (CONFIG_PROC_FS) [Y/n/?] **Press Enter**
This is a virtual file system providing information about the status of the system. Several programs depend on this, so everyone should say Y here.

/dev/pts file system for Unix98 PTYs (CONFIG_DEVPTS_FS) [Y/n/?] **Press Enter**
This option allows us to get a virtual file system which can be mounted on /dev/pts with "mount -t devpts". This, together with the pseudo terminal master multiplexer /dev/ptmx, is used for pseudo terminal support as described in The Open Group's Unix98 standard. Again, everyone should say Y here.

ROM file system support (CONFIG_ROMFS_FS) [N/y/?] **Press Enter**
This is a very small read-only file system mainly intended for initial ram disks of installation disks, but it could be used for other read-only media as well. Most people will simply say N to this question. If you want to run SCSI system on modularized kernel, you should say Y here.

Second extended fs support (CONFIG_EXT2_FS) [Y/n/?] **Press Enter**
This is the de facto standard Linux file system (method to organize files on a storage device) for hard disks and you must say Y to this question.

System V/Xenix/V7/Coherent file system support (CONFIG_SYSV_FS) [N/y/?] **Press Enter**
SCO, Xenix and Coherent are commercial Unix systems for Intel machines, and Version 7 was used on the DEC PDP-11. Saying Y here would allow you to read from their floppies and hard disk partitions. Most people will say N to this question.

UDF file system support (read only) (CONFIG_UDF_FS) [N/y/?] **Press Enter**
This is the new file system used on some CD-ROMs and DVD's. Say Y if you intend to mount DVD discs or CD-RW's written in packet mode, or if written to by other UDF utilities, such as DirectCD. Only enable this option if you have some such need.

UFS file system support (read only) (CONFIG_UFS_FS) [N/y/?] **Press Enter**
BSD and derivative versions of Unix (such as SunOS, FreeBSD, NetBSD, OpenBSD and NeXTstep) use a file system called UFS. Some System V Unixes can create and mount hard disk partitions and diskettes using this file system as well. Saying Y here will allow you to read from these partitions. Most users will say N here.

*
*** Network File Systems**
* Coda file system support (advanced network fs) (CONFIG_CODA_FS) [N/y/?] **Press Enter**
Coda is an advanced network file system, similar to NFS in that it enables you to mount file systems of a remote server and access them with regular Unix commands as if they were sitting on your hard disk. Enable this option only if you need it otherwise disable it.

NFS file system support (CONFIG_NFS_FS) [Y/n/?] **n**
If you are connected to some other (usually local) Unix computer (using SLIP, PLIP, PPP or Ethernet) and want to mount files residing on that computer (the NFS server) using the **N**etwork **F**ile **S**haring protocol, say Y here.

NFS server support (CONFIG_NFSD) [Y/n/?] **n**
If you want your Linux box to act as an NFS *server*, so that other computers on your local network which support NFS can access certain directories on your box transparently, you have two options: you can use the self-contained user space program nfsd, in which case you should say N here, or you can say Y and use the kernel based NFS server. The advantage of the kernel based solution is that it is faster. Finally, if you don't want to support NFS server at all, simply say N here.

SMB file system support (to mount Windows shares etc.) (CONFIG_SMB_FS) [N/y/?] **Press Enter**
SMB (**S**erver **M**essage **B**lock) is the protocol **W**indows **f**or **W**orkgroups (WfW), Windows 95/98, Windows NT, 2000, XP and OS/2 Lan Manager use to share files and printers over local networks. Saying Y here allows you to mount their file systems (often called "shares" in this context) and access them just like any other Unix directory. Enable this option only if you need it. In most cases the answer to this question will be N even if you install Samba on your system.

NCP file system support (to mount NetWare volumes) (CONFIG_NCP_FS) [N/y/?] **Press Enter**
NCP (**N**etWare **C**ore **P**rotocol) is a protocol that runs over IPX and is used by Novell NetWare clients to talk to file servers. It is to IPX what NFS is to TCP/IP, if that helps. Saying Y here allows you to mount NetWare file server volumes and to access them just like any other Unix directory. Enable this option only if you need it. In most cases the answer to this question will be N. You do not have to say Y here if you want your Linux box to act as a file *server* for Novell NetWare clients.

```
*
* Partition Types
* Advanced partition selection (CONFIG_PARTITION_ADVANCED) [N/y/?] Press Enter
```
This option allows us to enable the use of hard disks under Linux which were partitioned under an operating system running on a different architecture than the Linux system. Note that the answer to this question won't directly affect the kernel: saying `N` will just cause the configuration to skip all the questions about foreign partitioning schemes.

```
*
* Console drivers
* VGA text console (CONFIG_VGA_CONSOLE) [Y/n/?] Press Enter
```
This option allows us to use Linux in text mode through a display that complies with the generic VGA standard. Virtually everyone wants that. Everyone should say `Y` here.

```
Video mode selection support (CONFIG_VIDEO_SELECT) [N/y/?] Press Enter
```
This option allows us to enable support for text mode selection on kernel startup. In general we don't need to enable this option. Say `N` here and read the file `Documentation/svga.txt` for more information about the Video mode selection support if you are curious.

```
*
* Sound
* Sound card support (CONFIG_SOUND) [Y/n/?] n
```
This option allows us to enable sound support under Linux. If you have a sound card in your computer, then say `Y` here and select from the available list of sound card the one that you have. If you run Linux as a workstation, you may need to say `Y` here, if you run Linux as a server, you really don't need to enable this option and can safety say `N` here.

```
*
* USB support
* Support for USB (CONFIG_USB) [Y/n/?] n
```
This option allows us to enable `USB` support under Linux. If your computer has a `USB` port and you want to use `USB` devices, then you have to say `Y` here. For servers you really don't need to say `Y` here and can safety say `N`.

```
*
* Kernel hacking
* Kernel debugging (CONFIG_DEBUG_KERNEL) [N/y/?] Press Enter
```
You have to say `Y` here only if you are developing drivers or trying to debug and identify kernel problems. Most users will simply say `N` here.

```
*
* Grsecurity
*
Grsecurity (CONFIG_GRKERNSEC) [N/y/?] y
```
This option allows us to enable `Grsecurity` support under Linux. If you say `Y` to this option, you will be able to configure many `Grsecurity` features that will enhance the security of your Linux system in many ways. This option is available ONLY if you have patched your Linux kernel with the `Grsecurity` patch as discussed previously in this chapter. For best security of your Linux server say `Y` here.

```
Security level (Low, Medium, High, Customized) [Customized] Press Enter
```
This `Grsecurity` option allows us to choose between three predefined `Grsecurity` configurations or to customize the `Grsecurity` configuration as we want. For better control of what the patch does and what we may or may not need, we chose to have full control about what should or shouldn't be enable with `Grsecurity` by pressing the `[Enter]` key to accept the default choice of `[Customized]`, which let us see all available security features and chose only the one we need for our server and kernel security setup.

```
*
* Buffer Overflow Protection
*
Openwall non-executable stack (CONFIG_GRKERNSEC_STACK) [N/y/?] y
```
This Grsecurity option allows us to enable the non-executable stack protection on the system. If you say Y here, your system will not allow execution of code on the stack, making buffer overflow exploitation more difficult. It's a good idea to say Y here.

```
Gcc trampoline support (CONFIG_GRKERNSEC_STACK_GCC) [N/y/?] Press Enter
```
This Grsecurity option allows us to support trampoline code along with the above stack protection. Trampolining is an action to use the ability of the stack to contain executable code, which can improve program efficiency in a few cases. Since few programs and some version of GCC use and need 'trampolining', it is preferable to NOT enable this option to avoid break of some program on the system. Say N here by pressing the [Enter] key.

```
Read-only kernel memory (CONFIG_GRKERNSEC_KMEM) [N/y/?] y
```
This Grsecurity option allows us to enable the read-only kernel memory protection on the system. If you say Y here, root will not be able to modify the contents of kernel memory. It's a good idea to say Y here. If you are building a Monolithic Kernel, then the ability of an attacker to insert foreign code into a running kernel is completely removed. Yes, another good idea to build a Monolithic Kernel instead of a Modularized Kernel. Say Y here.

```
*
* Access Control Lists
*
Grsecurity ACL system (CONFIG_GRKERNSEC_ACL) [N/y/?] y
```
This Grsecurity option allows us to enable the **A**ccess **C**ontrol **L**ist system (ACL) for Grsecurity. It's a good idea to say Y here. ACL allows us to better control what program, files, etc on the system are allowed to do. We use it to apply a security policy that will work for the entire system. You can install and run Grsecurity without ACL but it is recommended for optimum security to enable this feature and use it. Once properly implemented, it will become impossible for a cracker to access and damage your Linux server. Personally, with Grsecurity ACL, I don't know how someone could break into a Linux system. Say Y here.

```
ACL Debugging Messages (CONFIG_GR_DEBUG) [N/y/?] y
```
This Grsecurity option allows the Grsecurity ACL system to print debugging messages as an aid to finding problems in your ACL sets. It's really a good idea to say Y here since it can become very difficult to debug problems with ACL if this option is not enable. Say Y here.

```
Extra ACL Debugging Messages (CONFIG_GR_SUPERDEBUG) [N/y/?] Press Enter
```
This Grsecurity option allows us to enable additional debugging messages that can help in finding problems in ACL sets or to gain a better understanding of the internal workings of the ACL system. In many cases, it is not necessary to enable this additional debugging messages feature for ACL and we will say N here.

```
Denied capability logging (CONFIG_GRKERNSEC_ACL_CAPLOG) [N/y/?] y
```
This Grsecurity option allows us to enable the denied capability logging support protection with ACL. If you say Y here, logs will be produced when a root-owned process does not have a needed capability raised in his set. This can help to debug the ACL of Grsecurity again. It's a good idea to say Y here.

```
Path to gradm (CONFIG_GRADM_PATH) [/sbin/gradm] Press Enter
```
This Grsecurity option allows us to specify the path of the gradm binary installed on the system. gradm is the binary program used to manage Grsecurity ACL on the server. You have to download and install it to be able to use ACL on your server. Please read the next chapter in this book to get information about gradm and how to setup Grsecurity ACL on your system. The default path for gradm as shown above is correct and we press the [Enter] key to accept the default value for the path.

```
Maximum tries before password lockout (CONFIG_GR_MAXTRIES) [3] 2
```
Once the gradm program is installed on your server to control and manage Grsecurity ACL, you will have to create a password for gradm to work. This option allows us to specify the number of times a user can attempt to authorize them selves with the Grsecurity ACL system. Here we change the default value of 3 to become 2, meaning that users are allowed to authorize themselves with the Grsecurity ACL system twice.

```
Time to wait after max password tries, in seconds (CONFIG_GR_TIMEOUT) [30]
Press Enter
```
This option specifies the time the user must wait after attempting to authorize to the ACL system with the maximum number of invalid passwords. Just press [Enter] to accept the default entry.

```
*
* Filesystem Protections
*
Proc restrictions (CONFIG_GRKERNSEC_PROC) [N/y/?] y
```
This Grsecurity option allows us to enable the proc restrictions protection on the system. If you say Y here, the permissions of the /proc file system will be altered to enhance system security and privacy. It's a very good idea to say Y here.

```
Restrict to user only (CONFIG_GRKERNSEC_PROC_USER) [N/y/?] y
```
This Grsecurity option allows us to enable restrict to user only protection on the system. If you say Y here, non-root users will only be able to view their own processes, restricts them from viewing network-related information, viewing kernel symbol and module information. It's a very good idea to say Y here.

```
Additional restrictions (CONFIG_GRKERNSEC_PROC_ADD) [N/y/?] y
```
This Grsecurity option allows us to enable additional restrictions protection on the system. If you say Y here, additional restrictions will be placed on the /proc file system that keep normal users from viewing cpu and device information. Again, it's a good idea to say Y here.

```
Linking restrictions (CONFIG_GRKERNSEC_LINK) [N/y/?] y
```
This Grsecurity option allows us to enable the linking restrictions protection on the system. If you say Y here, /tmp race exploits will be prevented, since users will no longer be able to follow symlinks owned by other users in world-writeable +t directories, unless the owner of the symlink is the owner of the directory. Users will also not be able to hard link to files they do not own. It's really a good idea to say Y here.

```
FIFO restrictions (CONFIG_GRKERNSEC_FIFO) [N/y/?] y
```
This Grsecurity option allows us to enable FIFO restrictions protection on the system. If you say Y here, users will not be able to write to FIFOs they don't own in world-writeable +t directories, unless the owner of the FIFO is the same owner of the directory it's held in. It's a good idea to say Y here.

`Secure file descriptors (CONFIG_GRKERNSEC_FD) [N/y/?]` **y**
This `Grsecurity` option allows us to enable secure file descriptors protection on the system. If you say `Y` here, binaries will be protected from data spoofing attacks. It's a very good idea to say `Y` here.

`Chroot jail restrictions (CONFIG_GRKERNSEC_CHROOT) [N/y/?]` **y**
This `Grsecurity` option allows us to enable chroot jail restrictions protection on the system. If you say `Y` here, you will be able to choose several options that will make breaking out of a chrooted jail much more difficult. It's a very good idea to say `Y` here.

`Restricted signals (CONFIG_GRKERNSEC_CHROOT_SIG) [N/y/?]` **y**
This `Grsecurity` option allows us to enable the restricted signals protection on the system. If you say `Y` here, processes inside a chroot will not be able to send signals outside of the chroot. It's a good idea to say `Y` here.

`Deny mounts (CONFIG_GRKERNSEC_CHROOT_MOUNT) [N/y/?]` **y**
This `Grsecurity` option allows us to enable deny mounts protection on the system. If you say `Y` here, processes inside a chroot will not be able to mount or remount file systems. It's a good idea to say `Y` here.

`Deny double-chroots (CONFIG_GRKERNSEC_CHROOT_DOUBLE) [N/y/?]` **y**
This `Grsecurity` option allows us to enable the deny double-chroot protection on the system. If you say `Y` here, processes inside a chroot will not be able to chroot again. This is a widely used method of breaking out of a chroot jail and should not be allowed. It's a good idea to say `Y` here.

`Enforce chdir("/") on all chroots (CONFIG_GRKERNSEC_CHROOT_CHDIR) [N/y/?]` **y**
This `Grsecurity` option allows us to enable the enforce `chdir("/")` on all chroots protection on the system. If you say `Y` here, the current working directory of all newly-chrooted applications will be set to the root directory of the chroot. It's a good idea to say `Y` here.

`Deny (f)chmod +s (CONFIG_GRKERNSEC_CHROOT_CHMOD) [N/y/?]` **y**
This `Grsecurity` option allows us to enable the deny `(f)chmod +s` protection on the system. If you say `Y` here, processes inside a chroot will not be able to `chmod` or `fchmod` files to make them have `suid` or `sgid` bits. It's a really good idea to say `Y` here.

`Deny mknod (CONFIG_GRKERNSEC_CHROOT_MKNOD) [N/y/?]` **y**
This `Grsecurity` option allows us to enable the deny mknod protection on the system. If you say `Y` here, processes inside a chroot will not be allowed to `mknod` (create device on the system). It's a good idea to say `Y` here, unless you run into software incompatibilities.

`Deny ptraces (CONFIG_GRKERNSEC_CHROOT_PTRACE) [N/y/?]` **y**
This `Grsecurity` option allows us to enable the deny ptraces protection on the system. If you say `Y` here, processes inside a chroot will not be able to ptrace other processes. It's a good idea to say `Y` here.

`Restrict priority changes (CONFIG_GRKERNSEC_CHROOT_NICE) [N/y/?]` **y**
This `Grsecurity` option allows us to enable the restrict priority changes protection on the system. If you say `Y` here, processes inside a chroot will not be able to raise the priority of processes in the chroot, or alter the priority of processes outside the chroot. It's a good idea to say `Y` here.

```
Capability restrictions within chroot (CONFIG_GRKERNSEC_CHROOT_CAPS) [N/y/?]
```
Press Enter

This Grsecurity option allows us to enable the capability restrictions within chroot protection on the system. If you say Y here, the capabilities on all root processes within a chroot jail will be lowered to stop module insertion, raw i/o, system and net admin tasks, rebooting the system, modifying immutable files, and changing the system time. This option can break some applications on the server and we disable it by answering to the question by N.

```
Secure keymap loading (CONFIG_GRKERNSEC_KBMAP) [N/y/?] y
```
This Grsecurity option allows us to enable the secure keymap loading protection on the system. If you say Y here, KDSKBENT and KDSKBSENT ioctl calls being called by unprivileged users will be denied. This means that unprivileged users with access to the console will NOT be able to modify keyboard bindings. It's a good idea to say Y here.

```
*
* Kernel Auditing
*
Single group for auditing (CONFIG_GRKERNSEC_AUDIT_GROUP) [N/y/?] Press Enter
```
This Grsecurity option allows us to enable single group for auditing protection on the system. If you say Y here, the exec, chdir, (un)mount, and ipc logging features of Grsecurity will only operate on a group you specify. By default Grsecurity produces a large amount of logs from the entire system on a production server; we don't really need the entire auditing feature provided by Grsecurity even on specified group. Therefore we simply say N to this question and enable later in this Grsecurity kernel configuration the auditing log that we really need for production servers.

```
Exec logging (CONFIG_GRKERNSEC_EXECLOG) [N/y/?] Press Enter
```
This Grsecurity option allows us to enable the exec logging protection on the system. If you say Y here, execution of any program on the server will be logged to syslog. This option when enabled will produce a LOT of logs, especially on an active system. Therefore, we don't recommend you to enable this security option. If you are those people that like to spend all their time reading log files, you could enable this option but be aware that it will take you a day to read all the logs generated by this option on a production server. Be reasonable, and say N here.

```
Log execs within chroot (CONFIG_GRKERNSEC_CHROOT_EXECLOG) [N/y/?] y
```
This Grsecurity option allows us to enable the log execs within chroot protection on the system. If you say Y here, all executions of any program inside a chroot jail environment will be logged to syslog. Contrary to the previous option that logs execution of any program on the system, this option ONLY logs the execution of program inside a chroot jail. In general, services running in chroot jail environment are limited, meaning that your log file will not become too BIG to read; therefore we can say Y here to enable this security option on the server.

```
Chdir logging (CONFIG_GRKERNSEC_AUDIT_CHDIR) [N/y/?] Press Enter
```
This Grsecurity option allows us to enable the chdir logging protection on the system. If you say Y here, all chdir() calls will be logged. Chdir() calls is when you navigate via your console on your server by using the 'cd' command. Imagine how many times you use the 'cd' command on your server when you performing some administration tasks. I recommend you to NOT enable this option if you want to avoid some BIG log files to read again. Say N here.

```
(Un)Mount logging (CONFIG_GRKERNSEC_AUDIT_MOUNT) [N/y/?] Press Enter
```
This Grsecurity option allows us to enable the (Un)Mount logging protection on the system. If you say Y here, all mounts and unmounts calls will be logged to syslog. This means that each time you mount or unmount drive on your server; the action will be logged to syslog. You can enable this security option if you like, but personally, I prefer to disable it. Say N here.

IPC logging (CONFIG_GRKERNSEC_AUDIT_IPC) [N/y/?] **y**
This Grsecurity option allows us to enable the IPC logging protection on the system. If you say Y here, creation and removal of message queues, semaphores, and shared memory will be logged. It's a good idea to say Y here.

Ptrace logging (CONFIG_GRKERNSEC_AUDIT_PTRACE) [N/y/?] **Press Enter**
This Grsecurity option allows us to enable the ptrace logging protection on the system. If you say Y here, all successful ptraces will be logged. Ptraces are special operations performed when programs like strace or gdb are run. In general we never install programs like strace and gdb on a production server. These programs are only required on development server to debug software. If you don't have these kinds of programs installed on your server, you can safety say N here, otherwise say Y to this security option.

Signal logging (CONFIG_GRKERNSEC_SIGNAL) [N/y/?] **y**
This Grsecurity option allows us to enable the signal logging protection on the system. If you say Y here, certain important signals will be logged, such as SIGSEGV that will inform you of when an error in a program occurred, which in some cases could mean a possible exploit attempt. It's a good idea to say Y here.

Fork failure logging (CONFIG_GRKERNSEC_FORKFAIL) [N/y/?] **y**
This Grsecurity option allows us to enable the fork failure logging protection on the system. If you say Y here, all failed fork() attempts will be logged. This could suggest a fork bomb, or someone attempting to overstep their process limit. It's a good idea to say Y here.

Set*id logging (CONFIG_GRKERNSEC_SUID) [N/y/?] **Press Enter**
This Grsecurity option allows us to enable the set*id logging protection on the system. If you say Y here, all set*id() calls will be logged. Enabling this security option could produce a lot of logs on an active system that run some services that use set*id() calls to operate. Mailman, Exim, Sendmail are known to software that uses many set*id() calls. Also, we already have other security programs doing the same job like sXid, therefore we don't really need to enable this option. It is your's to decide if you really need it or not. Personally, I disable this option by saying N here and use sXid to achieve the same result.

Log set*ids to root (CONFIG_GRKERNSEC_SUID_ROOT) [N/y/?] **y**
This Grsecurity option allows us to enable the log set*ids to root protection on the system. If you say Y here, only set*id() calls where a user is changing to the GID or UID of the root user will be logged. Such information could be useful when detecting a possible intrusion attempt. This option will produce smaller logs than logging all calls; therefore it's a good idea to say Y here.

Time change logging (CONFIG_GRKERNSEC_TIME) [N/y/?] **y**
This Grsecurity option allows us to enable the time change logging protection on the system. If you say Y here, any changes of the system clock will be logged. It's a good idea to say Y here.

*
*** Executable Protections**
*
Exec process limiting (CONFIG_GRKERNSEC_EXECVE) [N/y/?] **y**
This Grsecurity option allows us to enable the exec process limiting protection on the system. If you say Y here, users with a resource limit on processes will have the value checked during execve() calls (execution of program). It's really a good idea to say Y here.

`Dmesg(8) restriction (CONFIG_GRKERNSEC_DMESG) [N/y/?]` **y**
This `Grsecurity` option allows us to enable the dmesg restriction protection on the system. If you say `Y` here, non-root users will not be able to use `dmesg(8)` to view up to the last 4kb of messages in the kernel's log buffer. Again, it's really a good idea to say `Y` here.

`Randomized PIDs (CONFIG_GRKERNSEC_RANDPID) [N/y/?]` **y**
This `Grsecurity` option allows us to enable the randomized PIDs protection on the system. If you say `Y` here, all PIDs created on the system will be pseudo-randomly generated. This is extremely effective to disallow an attacker from guessing pids of daemons, etc. It's a good idea to say `Y` here.

`Altered default IPC permissions (CONFIG_GRKERNSEC_IPC) [N/y/?]` **Press Enter**
This `Grsecurity` option allows us to enable the altered default IPC permissions protection on the system. If you say `Y` here, the default permissions for `IPC` objects will be set based on the file system umask of the user creating the object. This is a good security feature but unfortunately, it is known to break software like Apache. Therefore we say `N` here.

`Limit uid/gid changes to root (CONFIG_GRKERNSEC_TTYROOT) [N/y/?]` **y**
This `Grsecurity` option allows us to enable the limit UID/GID changes to root protection on the system. If you say `Y` here, you will be able choose from three options that will allow you to restrict access to the root account by console type. Therefore we say `Y` here.

`Deny physical consoles (tty) (CONFIG_GRKERNSEC_TTYROOT_PHYS) [N/y/?]` **Press Enter**
This `Grsecurity` option allows us to enable the deny physical consoles (tty) protection on the system. If you say `Y` here, access to root from physical consoles will be denied. This is only recommended for rare cases where you will never need to be physically at the machine. For most of us, this is not the case and we have to say `N` here.

`Deny serial consoles (ttyS) (CONFIG_GRKERNSEC_TTYROOT_SERIAL) [N/y/?]` **y**
This `Grsecurity` option allows us to enable deny serial consoles (ttyS) protection on the system. If you say `Y` here, access to root from serial consoles will be denied. Most people can say `Y` here, since most don't use serial devices for their console access.

`Deny pseudo consoles (pty) (CONFIG_GRKERNSEC_TTYROOT_PSEUDO) [N/y/?]` **Press Enter**
This `Grsecurity` option allows us to enable the deny pseudo consoles (pty) protection on the system. If you say `Y` here, access to root from pseudo consoles will be denied. Pseudo consoles include consoles from `telnet`, `ssh`, or any other kind of interactive shell initiated from the network. In general, most of us use at least `SSH` to make a remote connection. Therefore we have to say `N` here, if we want to continue to use `SSH` for secure remote connection.

`Fork-bomb protection (CONFIG_GRKERNSEC_FORKBOMB) [N/y/?]` **y**
This `Grsecurity` option allows us to enable fork-bomb protection on the system. If you say `Y` here, you will be able to configure a group to add to users on your system that you want to be unable to fork-bomb the system. It's a good idea to say `Y` here and chose in the next security option the GID to run this protection with.

`GID for restricted users (CONFIG_GRKERNSEC_FORKBOMB_GID) [1006]` **Press Enter**
This `Grsecurity` option allows us to enable the GID for restricted users' protection on the system. Here we have to enter a GID number as the value, the default value is `1006` and we can keep it by pressing the `[Enter]` key. It is important to note that this GID should be added to any user for which the feature should be activated. See the next chapter of this book for more information about the procedures to follow. At this time you only need to accept the default value.

`Forks allowed per second (CONFIG_GRKERNSEC_FORKBOMB_SEC) [40]` **Press Enter**
This `Grsecurity` option allows us to enable the forks allowed per second protection on the system. This option is a continuation of the above forks options, here we have to enter the maximum number of forks allowed per second, and the default setting should be fine for most users.

`Maximum processes allowed (CONFIG_GRKERNSEC_FORKBOMB_MAX) [20]` **35**
This `Grsecurity` option allows us to enable the maximum processes allowed on the system. Here we have to enter the maximum number of processes users in the fork-bomb protected group can run. We change the default value of `20` to become `35`. `35` is the number you have set into the `/etc/security/limit.conf` file. Please see what is set into your `limit.conf` file and report it here. In general, `35` is a good value to go with.

`Trusted path execution (CONFIG_GRKERNSEC_TPE) [N/y/?]` **y**
This `Grsecurity` option allows us to enable trusted path execution protection on the system. If you say `Y` here, you will be able to choose a GID to add to the supplementary groups of users you want to mark as "untrusted." These users will not be able to execute any files that are not in root-owned directories writeable only by root. It's a good idea to say `Y` here.

`Glibc protection (CONFIG_GRKERNSEC_TPE_GLIBC) [N/y/?]` **y**
This `Grsecurity` option allows us to enable the glibc protection on the system. If you say `Y` here, all non-root users executing any files while `glibc` specific environment variables such as `LD_PRELOAD` are set, will have their environment cleared of these variables, since they could be used to evade the trusted path execution protection. It also protects against evasion through executing the dynamic linker to run a rogue binary. It is highly recommended you say `Y` here.

`Partially restrict non-root users (CONFIG_GRKERNSEC_TPE_ALL) [N/y/?]` **y**
This `Grsecurity` option allows us to enable the partially restrict non-root users protection on the system. If you say `Y` here, All non-root users other than the ones in the group specified in the main `TPE` option will only be allowed to execute files in directories they own that are not group or world-writeable, or in directories owned by root and writeable only by root. It's a good idea to say `Y` here.

`GID for untrusted users: (CONFIG_GRKERNSEC_TPE_GID) [1005]` **Press Enter**
This `Grsecurity` option allows us to enable the GID for untrusted user's protection on the system. Here we have to enter a GID number as the value, the default value is `1005` and we can keep it by pressing the `[Enter]` key. It is important to note that this GID should be added to any user for which the feature should be activated. See the next chapter of this book for more information about the procedures to follow. At this time you only need to accept the default value.

`Restricted ptrace (CONFIG_GRKERNSEC_PTRACE) [N/y/?]` **y**
This `Grsecurity` option allows us to enable the restricted ptrace protection on the system. If you say `Y` here, no one but root will be able to ptrace processes. Tracing syscalls inside the kernel will also be disabled. It's a good idea to say `Y` here.

`Allow ptrace for group (CONFIG_GRKERNSEC_PTRACE_GROUP) [N/y/?]` **Press Enter**
This `Grsecurity` option allows us to enable the allow ptrace for group protection on the system. If you say `Y` here, you will be able to choose a GID of users will be able to `ptrace`. Remember that `ptraces` are special operations performed when programs like `strace` or `gdb` are run for debugging purpose. Since these kind of program should in general be run on development server and by a root user only, we can safety say `N` here.

```
*
* Network Protections
*
Randomized IP IDs (CONFIG_GRKERNSEC_RANDID) [N/y/?] y
```

This Grsecurity option allows us to enable the allow randomized IP IDs protection on the system. If you say Y here, the entire ID field on all outgoing packets will be randomized. This hinders OS fingerprinters and keeps your machine from being used as a bounce for an untraceable portscan. It's a good idea to say Y here.

```
Randomized TCP source ports (CONFIG_GRKERNSEC_RANDSRC) [N/y/?] y
```

This Grsecurity option allows us to enable the randomized TCP source ports protection on the system. If you say Y here, situations where a source port is generated on the fly for the TCP protocol (ie. with connect()) will be altered so that the source port is generated at random, instead of a simple incrementing algorithm. It's a good idea to say Y here.

```
Randomized RPC XIDs (CONFIG_GRKERNSEC_RANDRPC) [N/y/?] y
```

This Grsecurity option allows us to enable the randomized RPC XIDs protection on the system. If you say Y here, the method of determining XIDs for RPC requests will be randomized, instead of using linux's default behavior of simply incrementing the XID. This allows us to have a more secure RPC connection on the system. It's a good idea to say Y here.

```
Altered Ping IDs (CONFIG_GRKERNSEC_RANDPING) [N/y/?] y
```

This Grsecurity option allows us to enable the altered Ping IDs protection on the system. If you say Y here, the way Linux handles echo replies will be changed so that the reply uses an ID equal to the ID of the echo request. This will help in confusing OS detection. It's a good idea to say Y here.

```
Randomized TTL (CONFIG_GRKERNSEC_RANDTTL) [N/y/?] y
```

This Grsecurity option allows us to enable the randomized TTL protection on the system. If you say Y here, your TTL (**t**ime **t**o **l**ive) for packets will be set at random, with a base of the sysctl ttl default, to further confuse OS detection. It's a good idea to say Y here.

```
Socket restrictions (CONFIG_GRKERNSEC_SOCKET) [N/y/?] y
```

This Grsecurity option allows us to enable the socket restrictions protection on the system. If you say Y here, you will be able to choose from three options related to socket protection on the server. From these three available security options, you'll have to choose the one that best fits your requirements. Therefore, it's a good idea to say Y here since we have to choose one of the three available security options for our needs.

```
Deny any sockets to group (CONFIG_GRKERNSEC_SOCKET_ALL) [N/y/?] y
```

This Grsecurity option allows us to enable the socket restrictions protection on the system. If you say Y here, you will be able to choose a GID of whose users will be unable to connect to other hosts from your machine or run server applications from your machine. This is the security option that we'll choose. Say Y here.

```
GID to deny all sockets for: (CONFIG_GRKERNSEC_SOCKET_ALL_GID) [1004] Press
Enter
```

This Grsecurity option allows us to enable the GID to deny all sockets protection on the system. Here we have to enter a GID number as the value, the default value is 1004 and we can keep it by pressing the [Enter] key. It is important to note that this GID should be added to any user for which the feature should be activated. See the next chapter of this book for more information about the procedures to follow. At this time you only need to accept the default value.

```
Deny client sockets to group (CONFIG_GRKERNSEC_SOCKET_CLIENT) [N/y/?] Press
Enter
```

This Grsecurity option allows us to enable the deny client sockets to group protection on the system. If you say Y here, you will be able to choose a GID of whose users will be unable to connect to other hosts from your machine, but will be able to run servers. We have already chosen the above option that enables socket protection on both ways (users cannot connect to other hosts or run server applications from our machine), therefore we don't need to say Y here. Say N here.

```
Deny server sockets to group (CONFIG_GRKERNSEC_SOCKET_SERVER) [N/y/?] Press
Enter
```

This Grsecurity option allows us to enable the deny server sockets to group protection on the system. If you say Y here, you will be able to choose a GID of whose users will be unable to run server applications from your machine. As for the above option, we already have chosen the first option that enable socket protection on both way (users cannot connect to other hosts or run server applications from our machine), therefore we don't need to say Y here. Say N here.

```
*
* Sysctl support
*
Sysctl support (CONFIG_GRKERNSEC_SYSCTL) [N/y/?] Press Enter
```

This Grsecurity option allows us to enable Grsecurity sysctl support protection on the system. If you say Y here, you will be able to change the options that Grsecurity runs with at bootup, without having to recompile your kernel. You can echo values to files in /proc/sys/kernel/grsecurity to enable (1) or disable (0) various features. Enabling this option will reduce the effectiveness of the added security of the Grsecurity patch, therefore we say N here.

```
*
* Miscellaneous Features
*
Seconds in between log messages(minimum) (CONFIG_GRKERNSEC_FLOODTIME) [30]
Press Enter
```

This Grsecurity option allows us to enable the seconds in between log messages protection on the system. This option allows you to enforce the number of seconds between Grsecurity log messages. The default should be suitable for most people. Just press the [Enter] key here to accept the default value.

```
BSD-style coredumps (CONFIG_GRKERNSEC_COREDUMP) [N/y/?] y
```

This Grsecurity option allows us to enable the BSD-style coredumps protection on the system. If you say Y here, Linux will use a style similar to BSD for coredumps, core.processname. Not a security feature, just a useful one. Say Y here.

```
*** End of Linux kernel configuration.
*** Check the top-level Makefile for additional configuration.
*** Next, you must run 'make dep'.
```

Modularized kernel configuration

Building kernel with modules (`Modularized kernel`) has some advantages. It allows easy portability between different Linux systems, since you can choose and build different parts of the kernel as a module and load that segment of code on demand. Below we show you the configuration of `Modularized kernel`, which is to compile some needed codes and drivers as a module into the kernel by answering the different questions using **y**, **n** or **m**. As for the previous `Monolithic kernel` configuration, don't forget to only compile code that you need and use.

A new kernel is very specific to your computer hardware, in the `Modularized kernel` configuration part below; we assume the following hardware for our example. Of course you must change them to fit your system components.

1 Pentium-III 667 MHz (i686) processor
1 Motherboard Asus P3V4X Pro 133Mhz EIDE
1 Hard Disk Ultra ATA/100 EIDE
1 Chipset Apollo Pro133A
1 CD-ROM ATAPI IDE
1 Floppy Disk
2 Ethernet Cards 3COM 3c597 PCI 10/100
1 Mouse PS/2

If you don't want some options listed in the `Modularized kernel` configuration that I enable by default, answer **n** (for no) instead of **y** (for yes) or **m** (for modularized if possible) to the related questions. If you want some other options that I disable, then answer **y** or **m** instead of **n**.

```
rm -f include/asm
( cd include ; ln -sf asm-i386 asm)
/bin/sh scripts/Configure arch/i386/config.in
#
# Using defaults found in arch/i386/defconfig
#
*
* Code maturity level options
*
Prompt for development and/or incomplete code/drivers (CONFIG_EXPERIMENTAL) [N/y/?] Press
Enter
*
* Loadable module support
*
Enable loadable module support (CONFIG_MODULES) [Y/n/?] Press Enter
  Set version information on all module symbols (CONFIG_MODVERSIONS) [Y/n/?] n
  Kernel module loader (CONFIG_KMOD) [Y/n/?] Press Enter
*
* Processor type and features
*
Processor family (386, 486, 586/K5/5x86/6x86/6x86MX, Pentium-Classic, Pentium-MMX,
Pentium-Pro/Celeron/Pentium-II, Pentium-III/Celeron(Coppermine), Pentium-4, K6/K6-II/K6-
III, Athlon/Duron/K7, Elan, Crusoe, Winchip-C6, Winchip-2, Winchip-2A/Winchip-3,
CyrixIII/C3) [Pentium-III/Celeron(Coppermine)] Press Enter
Toshiba Laptop support (CONFIG_TOSHIBA) [N/y/m/?] Press Enter
Dell laptop support (CONFIG_I8K) [N/y/m/?] Press Enter
/dev/cpu/microcode - Intel IA32 CPU microcode support (CONFIG_MICROCODE) [N/y/m/?] m
/dev/cpu/*/msr - Model-specific register support (CONFIG_X86_MSR) [N/y/m/?] m
/dev/cpu/*/cpuid - CPU information support (CONFIG_X86_CPUID) [N/y/m/?] m
High Memory Support (off, 4GB, 64GB) [off] Press Enter
Math emulation (CONFIG_MATH_EMULATION) [N/y/?] Press Enter
MTRR (Memory Type Range Register) support (CONFIG_MTRR) [N/y/?] Press Enter
Symmetric multi-processing support (CONFIG_SMP) [Y/n/?] n
Local APIC support on uniprocessors (CONFIG_X86_UP_APIC) [N/y/?] (NEW) y
IO-APIC support on uniprocessors (CONFIG_X86_UP_IOAPIC) [N/y/?] (NEW) y
*
* General setup
```

```
*
Networking support (CONFIG_NET) [Y/n/?] Press Enter
PCI support (CONFIG_PCI) [Y/n/?] Press Enter
  PCI access mode (BIOS, Direct, Any) [Any] Press Enter
PCI device name database (CONFIG_PCI_NAMES) [Y/n/?] n
EISA support (CONFIG_EISA) [N/y/?] Press Enter
MCA support (CONFIG_MCA) [N/y/?] Press Enter
Support for hot-pluggable devices (CONFIG_HOTPLUG) [Y/n/?] n
System V IPC (CONFIG_SYSVIPC) [Y/n/?] Press Enter
BSD Process Accounting (CONFIG_BSD_PROCESS_ACCT) [N/y/?] Press Enter
Sysctl support (CONFIG_SYSCTL) [Y/n/?] Press Enter
Kernel core (/proc/kcore) format (ELF, A.OUT) [ELF] Press Enter
Kernel support for a.out binaries (CONFIG_BINFMT_AOUT) [Y/m/n/?] m
Kernel support for ELF binaries (CONFIG_BINFMT_ELF) [Y/m/n/?] Press Enter
Kernel support for MISC binaries (CONFIG_BINFMT_MISC) [Y/m/n/?] m
Power Management support (CONFIG_PM) [Y/n/?] n
*
* Memory Technology Devices (MTD)
*
Memory Technology Device (MTD) support (CONFIG_MTD) [N/y/m/?] Press Enter
*
* Parallel port support
*
Parallel port support (CONFIG_PARPORT) [N/y/m/?] Press Enter
*
* Plug and Play configuration
*
Plug and Play support (CONFIG_PNP) [Y/m/n/?] n
*
* Block devices
*
Normal PC floppy disk support (CONFIG_BLK_DEV_FD) [Y/m/n/?] Press Enter
XT hard disk support (CONFIG_BLK_DEV_XD) [N/y/m/?] Press Enter
Compaq SMART2 support (CONFIG_BLK_CPQ_DA) [N/y/m/?] Press Enter
Compaq Smart Array 5xxx support (CONFIG_BLK_CPQ_CISS_DA) [N/y/m/?] Press Enter
Mylex DAC960/DAC1100 PCI RAID Controller support (CONFIG_BLK_DEV_DAC960) [N/y/m/?] Press
Enter
Loopback device support (CONFIG_BLK_DEV_LOOP) [N/y/m/?] Press Enter
Network block device support (CONFIG_BLK_DEV_NBD) [N/y/m/?] Press Enter
RAM disk support (CONFIG_BLK_DEV_RAM) [N/y/m/?] Press Enter
*
* Multi-device support (RAID and LVM)
*
Multiple devices driver support (RAID and LVM) (CONFIG_MD) [N/y/?] Press Enter
*
* Networking options
*
Packet socket (CONFIG_PACKET) [Y/m/n/?] Press Enter
  Packet socket: mmapped IO (CONFIG_PACKET_MMAP) [N/y/?] y
Netlink device emulation (CONFIG_NETLINK_DEV) [N/y/m/?] (NEW) m
Network packet filtering (replaces ipchains) (CONFIG_NETFILTER) [N/y/?] y
  Network packet filtering debugging (CONFIG_NETFILTER_DEBUG) [N/y/?] (NEW) y
Socket Filtering (CONFIG_FILTER) [N/y/?] Press Enter
Unix domain sockets (CONFIG_UNIX) [Y/m/n/?] Press Enter
TCP/IP networking (CONFIG_INET) [Y/n/?] Press Enter
  IP: multicasting (CONFIG_IP_MULTICAST) [Y/n/?] n
  IP: advanced router (CONFIG_IP_ADVANCED_ROUTER) [N/y/?] Press Enter
  IP: kernel level autoconfiguration (CONFIG_IP_PNP) [N/y/?] Press Enter
  IP: tunneling (CONFIG_NET_IPIP) [N/y/m/?] Press Enter
  IP: GRE tunnels over IP (CONFIG_NET_IPGRE) [N/y/m/?] Press Enter
  IP: TCP Explicit Congestion Notification support (CONFIG_INET_ECN) [N/y/?] Press Enter
  IP: TCP syncookie support (disabled default) (CONFIG_SYN_COOKIES) [N/y/?] y
*
*   IP: Netfilter Configuration
*
Connection tracking (required for masq/NAT) (CONFIG_IP_NF_CONNTRACK) [N/y/m/?] (NEW) m
  FTP protocol support (CONFIG_IP_NF_FTP) [N/m/?] (NEW) m
  IRC protocol support (CONFIG_IP_NF_IRC) [N/m/?] (NEW) m
IP tables support (required for filtering/masq/NAT) (CONFIG_IP_NF_IPTABLES) [N/y/m/?]
(NEW) m
  limit match support (CONFIG_IP_NF_MATCH_LIMIT) [N/m/?] (NEW) m
```

```
  MAC address match support (CONFIG_IP_NF_MATCH_MAC) [N/m/?] (NEW) m
  netfilter MARK match support (CONFIG_IP_NF_MATCH_MARK) [N/m/?] (NEW) m
  Multiple port match support (CONFIG_IP_NF_MATCH_MULTIPORT) [N/m/?] (NEW) m
  TOS match support (CONFIG_IP_NF_MATCH_TOS) [N/m/?] (NEW) m
  AH/ESP match support (CONFIG_IP_NF_MATCH_AH_ESP) [N/m/?] (NEW) m
  LENGTH match support (CONFIG_IP_NF_MATCH_LENGTH) [N/m/?] (NEW) m
  TTL match support (CONFIG_IP_NF_MATCH_TTL) [N/m/?] (NEW) m
  tcpmss match support (CONFIG_IP_NF_MATCH_TCPMSS) [N/m/?] (NEW) m
  Connection state match support (CONFIG_IP_NF_MATCH_STATE) [N/m/?] (NEW) m
  Packet filtering (CONFIG_IP_NF_FILTER) [N/m/?] (NEW) m
    REJECT target support (CONFIG_IP_NF_TARGET_REJECT) [N/m/?] (NEW) m
  Full NAT (CONFIG_IP_NF_NAT) [N/m/?] (NEW) m
    MASQUERADE target support (CONFIG_IP_NF_TARGET_MASQUERADE) [N/m/?] (NEW) m
    REDIRECT target support (CONFIG_IP_NF_TARGET_REDIRECT) [N/m/?] (NEW) m
  Packet mangling (CONFIG_IP_NF_MANGLE) [N/m/?] (NEW) m
    TOS target support (CONFIG_IP_NF_TARGET_TOS) [N/m/?] (NEW) m
    MARK target support (CONFIG_IP_NF_TARGET_MARK) [N/m/?] (NEW) m
  LOG target support (CONFIG_IP_NF_TARGET_LOG) [N/m/?] (NEW) m
  TCPMSS target support (CONFIG_IP_NF_TARGET_TCPMSS) [N/m/?] (NEW) m
ipchains (2.2-style) support (CONFIG_IP_NF_COMPAT_IPCHAINS) [N/y/m/?] (NEW) Press Enter
ipfwadm (2.0-style) support (CONFIG_IP_NF_COMPAT_IPFWADM) [N/y/m/?] (NEW) Press Enter
*
*
*
The IPX protocol (CONFIG_IPX) [N/y/m/?] Press Enter
Appletalk protocol support (CONFIG_ATALK) [N/y/m/?] Press Enter
DECnet Support (CONFIG_DECNET) [N/y/m/?] Press Enter
802.1d Ethernet Bridging (CONFIG_BRIDGE) [N/y/m/?] Press Enter
*
* QoS and/or fair queueing
*
QoS and/or fair queueing (CONFIG_NET_SCHED) [N/y/?] Press Enter
*
* Telephony Support
*
Linux telephony support (CONFIG_PHONE) [N/y/m/?] Press Enter
*
* ATA/IDE/MFM/RLL support
*
ATA/IDE/MFM/RLL support (CONFIG_IDE) [Y/m/n/?] Press Enter
*
* IDE, ATA and ATAPI Block devices
*
Enhanced IDE/MFM/RLL disk/cdrom/tape/floppy support (CONFIG_BLK_DEV_IDE) [Y/m/n/?] Press
Enter
*
* Please see Documentation/ide.txt for help/info on IDE drives
*
  Use old disk-only driver on primary interface (CONFIG_BLK_DEV_HD_IDE) [N/y/?] Press
Enter
  Include IDE/ATA-2 DISK support (CONFIG_BLK_DEV_IDEDISK) [Y/m/n/?] Press Enter
    Use multi-mode by default (CONFIG_IDEDISK_MULTI_MODE) [Y/n/?] n
  Include IDE/ATAPI CDROM support (CONFIG_BLK_DEV_IDECD) [Y/m/n/?] Press Enter
  Include IDE/ATAPI TAPE support (CONFIG_BLK_DEV_IDETAPE) [N/y/m/?] Press Enter
  Include IDE/ATAPI FLOPPY support (CONFIG_BLK_DEV_IDEFLOPPY) [N/y/m/?] Press Enter
  SCSI emulation support (CONFIG_BLK_DEV_IDESCSI) [N/y/m/?] Press Enter
*
* IDE chipset support/bugfixes
*
  CMD640 chipset bugfix/support (CONFIG_BLK_DEV_CMD640) [Y/n/?] n
  RZ1000 chipset bugfix/support (CONFIG_BLK_DEV_RZ1000) [Y/n/?] n
  Generic PCI IDE chipset support (CONFIG_BLK_DEV_IDEPCI) [Y/n/?] Press Enter
    Sharing PCI IDE interrupts support (CONFIG_IDEPCI_SHARE_IRQ) [Y/n/?] Press Enter
    Generic PCI bus-master DMA support (CONFIG_BLK_DEV_IDEDMA_PCI) [Y/n/?] Press Enter
    Boot off-board chipsets first support (CONFIG_BLK_DEV_OFFBOARD) [N/y/?] Press Enter
      Use PCI DMA by default when available (CONFIG_IDEDMA_PCI_AUTO) [Y/n/?] Press Enter
    AEC62XX chipset support (CONFIG_BLK_DEV_AEC62XX) [N/y/?] Press Enter
    ALI M15x3 chipset support (CONFIG_BLK_DEV_ALI15X3) [N/y/?] Press Enter
    AMD Viper support (CONFIG_BLK_DEV_AMD74XX) [N/y/?] Press Enter
    CMD64X chipset support (CONFIG_BLK_DEV_CMD64X) [N/y/?] Press Enter
    CY82C693 chipset support (CONFIG_BLK_DEV_CY82C693) [N/y/?] Press Enter
```

```
    Cyrix CS5530 MediaGX chipset support (CONFIG_BLK_DEV_CS5530) [N/y/?] Press Enter
    HPT34X chipset support (CONFIG_BLK_DEV_HPT34X) [N/y/?] Press Enter
    HPT366 chipset support (CONFIG_BLK_DEV_HPT366) [N/y/?] Press Enter
    Intel PIIXn chipsets support (CONFIG_BLK_DEV_PIIX) [Y/n/?] Press Enter
      PIIXn Tuning support (CONFIG_PIIX_TUNING) [Y/n/?] Press Enter
    NS87415 chipset support (EXPERIMENTAL) (CONFIG_BLK_DEV_NS87415) [N/y/?] Press Enter
    PROMISE PDC202{46|62|65|67|68} support (CONFIG_BLK_DEV_PDC202XX) [N/y/?] Press Enter
    ServerWorks OSB4/CSB5 chipsets support (CONFIG_BLK_DEV_SVWKS) [N/y/?] Press Enter
    SiS5513 chipset support (CONFIG_BLK_DEV_SIS5513) [N/y/?] Press Enter
    SLC90E66 chipset support (CONFIG_BLK_DEV_SLC90E66) [N/y/?] Press Enter
    Tekram TRM290 chipset support (EXPERIMENTAL) (CONFIG_BLK_DEV_TRM290) [N/y/?] Press
Enter
    VIA82CXXX chipset support (CONFIG_BLK_DEV_VIA82CXXX) [N/y/?] y
  Other IDE chipset support (CONFIG_IDE_CHIPSETS) [N/y/?] Press Enter
  IGNORE word93 Validation BITS (CONFIG_IDEDMA_IVB) [N/y/?] Press Enter
*
* SCSI support
*
SCSI support (CONFIG_SCSI) [Y/m/n/?] n
*
* Fusion MPT device support
*
*
* I2O device support
*
I2O support (CONFIG_I2O) [N/y/m/?] Press Enter
*
* Network device support
*
Network device support (CONFIG_NETDEVICES) [Y/n/?] Press Enter
*
* ARCnet devices
*
ARCnet support (CONFIG_ARCNET) [N/y/m/?] Press Enter
Dummy net driver support (CONFIG_DUMMY) [M/n/y/?] Press Enter
Bonding driver support (CONFIG_BONDING) [N/y/m/?] Press Enter
EQL (serial line load balancing) support (CONFIG_EQUALIZER) [N/y/m/?] Press Enter
Universal TUN/TAP device driver support (CONFIG_TUN) [N/y/m/?] Press Enter
*
* Ethernet (10 or 100Mbit)
*
Ethernet (10 or 100Mbit) (CONFIG_NET_ETHERNET) [Y/n/?] Press Enter
  Sun Happy Meal 10/100baseT support (CONFIG_HAPPYMEAL) [N/y/m/?] Press Enter
  Sun GEM support (CONFIG_SUNGEM) [N/y/m/?] Press Enter
  3COM cards (CONFIG_NET_VENDOR_3COM) [N/y/?] y
    3c501 "EtherLink" support (CONFIG_EL1) [N/y/m/?] (NEW) Press Enter
    3c503 "EtherLink II" support (CONFIG_EL2) [N/y/m/?] (NEW) Press Enter
    3c505 "EtherLink Plus" support (CONFIG_ELPLUS) [N/y/m/?] (NEW) Press Enter
    3c509/3c529 (MCA)/3c579 "EtherLink III" support (CONFIG_EL3) [N/y/m/?] (NEW) Press
Enter
    3c515 ISA "Fast EtherLink" (CONFIG_3C515) [N/y/m/?] (NEW) Press Enter
    3c590/3c900 series (592/595/597) "Vortex/Boomerang" support (CONFIG_VORTEX) [N/y/m/?]
(NEW) y
  AMD LANCE and PCnet (AT1500 and NE2100) support (CONFIG_LANCE) [N/y/m/?] Press Enter
  Western Digital/SMC cards (CONFIG_NET_VENDOR_SMC) [N/y/?] Press Enter
  Racal-Interlan (Micom) NI cards (CONFIG_NET_VENDOR_RACAL) [N/y/?] Press Enter
  DEPCA, DE10x, DE200, DE201, DE202, DE422 support (CONFIG_DEPCA) [N/y/m/?] Press Enter
  HP 10/100VG PCLAN (ISA, EISA, PCI) support (CONFIG_HP100) [N/y/m/?] Press Enter
  Other ISA cards (CONFIG_NET_ISA) [N/y/?] Press Enter
  EISA, VLB, PCI and on board controllers (CONFIG_NET_PCI) [Y/n/?] n
  Pocket and portable adapters (CONFIG_NET_POCKET) [N/y/?] Press Enter
*
* Ethernet (1000 Mbit)
*
Alteon AceNIC/3Com 3C985/NetGear GA620 Gigabit support (CONFIG_ACENIC) [N/y/m/?] Press
Enter
D-Link DL2000-based Gigabit Ethernet support (CONFIG_DL2K) [N/y/m/?] Press Enter
National Semiconduct DP83820 support (CONFIG_NS83820) [N/y/m/?] Press Enter
Packet Engines Hamachi GNIC-II support (CONFIG_HAMACHI) [N/y/m/?] Press Enter
SysKonnect SK-98xx support (CONFIG_SK98LIN) [N/y/m/?] Press Enter
FDDI driver support (CONFIG_FDDI) [N/y/?] Press Enter
```

```
PPP (point-to-point protocol) support (CONFIG_PPP) [N/y/m/?] Press Enter
SLIP (serial line) support (CONFIG_SLIP) [N/y/m/?] Press Enter
*
* Wireless LAN (non-hamradio)
*
Wireless LAN (non-hamradio) (CONFIG_NET_RADIO) [N/y/?] Press Enter
*
* Token Ring devices
*
Token Ring driver support (CONFIG_TR) [N/y/?] Press Enter
Fibre Channel driver support (CONFIG_NET_FC) [N/y/?] Press Enter
*
* Wan interfaces
*
Wan interfaces support (CONFIG_WAN) [N/y/?] Press Enter
*
* Amateur Radio support
*
Amateur Radio support (CONFIG_HAMRADIO) [N/y/?] Press Enter
*
* IrDA (infrared) support
*
IrDA subsystem support (CONFIG_IRDA) [N/y/m/?] Press Enter
*
* ISDN subsystem
*
ISDN support (CONFIG_ISDN) [N/y/m/?] Press Enter
*
* Old CD-ROM drivers (not SCSI, not IDE)
*
Support non-SCSI/IDE/ATAPI CDROM drives (CONFIG_CD_NO_IDESCSI) [N/y/?] Press Enter
*
* Input core support
*
Input core support (CONFIG_INPUT) [N/y/m/?] Press Enter
*
* Character devices
*
Virtual terminal (CONFIG_VT) [Y/n/?] Press Enter
  Support for console on virtual terminal (CONFIG_VT_CONSOLE) [Y/n/?] Press Enter
Standard/generic (8250/16550 and compatible UARTs) serial support (CONFIG_SERIAL)
[Y/m/n/?] Press Enter
  Support for console on serial port (CONFIG_SERIAL_CONSOLE) [N/y/?] Press Enter
Extended dumb serial driver options (CONFIG_SERIAL_EXTENDED) [N/y/?] Press Enter
Non-standard serial port support (CONFIG_SERIAL_NONSTANDARD) [N/y/?] Press Enter
Unix98 PTY support (CONFIG_UNIX98_PTYS) [Y/n/?] Press Enter
Maximum number of Unix98 PTYs in use (0-2048) (CONFIG_UNIX98_PTY_COUNT) [256] 128
*
* I2C support
*
I2C support (CONFIG_I2C) [N/y/m/?] Press Enter
*
* Mice
*
Bus Mouse Support (CONFIG_BUSMOUSE) [N/y/m/?] Press Enter
Mouse Support (not serial and bus mice) (CONFIG_MOUSE) [Y/m/n/?] n
*
* Joysticks
*
*
* Input core support is needed for gameports
*
*
* Input core support is needed for joysticks
*
QIC-02 tape support (CONFIG_QIC02_TAPE) [N/y/m/?] Press Enter
*
* Watchdog Cards
*
Watchdog Timer Support (CONFIG_WATCHDOG) [N/y/?] Press Enter
Intel i8x0 Random Number Generator support (CONFIG_INTEL_RNG) [N/y/m/?] Press Enter
```

```
/dev/nvram support (CONFIG_NVRAM) [N/y/m/?] Press Enter
Enhanced Real Time Clock Support (CONFIG_RTC) [N/y/m/?] Press Enter
Double Talk PC internal speech card support (CONFIG_DTLK) [N/y/m/?] Press Enter
Siemens R3964 line discipline (CONFIG_R3964) [N/y/m/?] Press Enter
Applicom intelligent fieldbus card support (CONFIG_APPLICOM) [N/y/m/?] Press Enter
*
* Ftape, the floppy tape device driver
*
Ftape (QIC-80/Travan) support (CONFIG_FTAPE) [N/y/m/?] Press Enter
/dev/agpgart (AGP Support) (CONFIG_AGP) [Y/m/n/?] n
Direct Rendering Manager (XFree86 DRI support) (CONFIG_DRM) [Y/n/?] n
ACP Modem (Mwave) support (CONFIG_MWAVE) [N/y/m/?] Press Enter
*
* Multimedia devices
*
Video For Linux (CONFIG_VIDEO_DEV) [N/y/m/?] Press Enter
*
* File systems
*
Quota support (CONFIG_QUOTA) [N/y/?] y
Kernel automounter support (CONFIG_AUTOFS_FS) [N/y/m/?] Press Enter
Kernel automounter version 4 support (also supports v3) (CONFIG_AUTOFS4_FS) [Y/m/n/?] n
Reiserfs support (CONFIG_REISERFS_FS) [N/y/m/?] Press Enter
Ext3 journalling file system support (EXPERIMENTAL) (CONFIG_EXT3_FS) [N/y/m/?] y
  JBD (ext3) debugging support (CONFIG_JBD_DEBUG) [N/y/?] y
DOS FAT fs support (CONFIG_FAT_FS) [N/y/m/?] m
  MSDOS fs support (CONFIG_MSDOS_FS) [N/y/m/?] m
  VFAT (Windows-95) fs support (CONFIG_VFAT_FS) [N/y/m/?] m
Compressed ROM file system support (CONFIG_CRAMFS) [N/y/m/?] Press Enter
Virtual memory file system support (former shm fs) (CONFIG_TMPFS) [Y/n/?] Press Enter
Simple RAM-based file system support (CONFIG_RAMFS) [N/y/m/?] Press Enter
ISO 9660 CDROM file system support (CONFIG_ISO9660_FS) [Y/m/n/?] m
  Microsoft Joliet CDROM extensions (CONFIG_JOLIET) [N/y/?] y
  Transparent decompression extension (CONFIG_ZISOFS) [N/y/?] Press Enter
Minix fs support (CONFIG_MINIX_FS) [N/y/m/?] Press Enter
FreeVxFS file system support (VERITAS VxFS(TM) compatible) (CONFIG_VXFS_FS) [N/y/m/?]
Press Enter
NTFS file system support (read only) (CONFIG_NTFS_FS) [N/y/m/?] Press Enter
OS/2 HPFS file system support (CONFIG_HPFS_FS) [N/y/m/?] Press Enter
/proc file system support (CONFIG_PROC_FS) [Y/n/?] Press Enter
/dev/pts file system for Unix98 PTYs (CONFIG_DEVPTS_FS) [Y/n/?] Press Enter
ROM file system support (CONFIG_ROMFS_FS) [N/y/m/?] Press Enter
Second extended fs support (CONFIG_EXT2_FS) [Y/m/n/?] Press Enter
System V/Xenix/V7/Coherent file system support (CONFIG_SYSV_FS) [N/y/m/?] Press Enter
UDF file system support (read only) (CONFIG_UDF_FS) [N/y/m/?] Press Enter
UFS file system support (read only) (CONFIG_UFS_FS) [N/y/m/?] Press Enter
*
* Network File Systems
*
Coda file system support (advanced network fs) (CONFIG_CODA_FS) [N/y/m/?] Press Enter
NFS file system support (CONFIG_NFS_FS) [Y/m/n/?] n
NFS server support (CONFIG_NFSD) [Y/m/n/?] n
SMB file system support (to mount Windows shares etc.) (CONFIG_SMB_FS) [N/y/m/?] Press
Enter
NCP file system support (to mount NetWare volumes) (CONFIG_NCP_FS) [N/y/m/?] Press Enter
*
* Partition Types
*
Advanced partition selection (CONFIG_PARTITION_ADVANCED) [N/y/?] Press Enter
*
* Native Language Support
*
Default NLS Option (CONFIG_NLS_DEFAULT) [iso8859-1] (NEW) Press Enter
Codepage 437 (United States, Canada) (CONFIG_NLS_CODEPAGE_437) [N/y/m/?] (NEW) Press
Enter
Codepage 737 (Greek) (CONFIG_NLS_CODEPAGE_737) [N/y/m/?] (NEW) Press Enter
Codepage 775 (Baltic Rim) (CONFIG_NLS_CODEPAGE_775) [N/y/m/?] (NEW) Press Enter
Codepage 850 (Europe) (CONFIG_NLS_CODEPAGE_850) [N/y/m/?] (NEW) Press Enter
Codepage 852 (Central/Eastern Europe) (CONFIG_NLS_CODEPAGE_852) [N/y/m/?] (NEW) Press
Enter
Codepage 855 (Cyrillic) (CONFIG_NLS_CODEPAGE_855) [N/y/m/?] (NEW) Press Enter
```

```
Codepage 857 (Turkish) (CONFIG_NLS_CODEPAGE_857) [N/y/m/?] (NEW) Press Enter
Codepage 860 (Portuguese) (CONFIG_NLS_CODEPAGE_860) [N/y/m/?] (NEW) Press Enter
Codepage 861 (Icelandic) (CONFIG_NLS_CODEPAGE_861) [N/y/m/?] (NEW) Press Enter
Codepage 862 (Hebrew) (CONFIG_NLS_CODEPAGE_862) [N/y/m/?] (NEW) Press Enter
Codepage 863 (Canadian French) (CONFIG_NLS_CODEPAGE_863) [N/y/m/?] (NEW) Press Enter
Codepage 864 (Arabic) (CONFIG_NLS_CODEPAGE_864) [N/y/m/?] (NEW) Press Enter
Codepage 865 (Norwegian, Danish) (CONFIG_NLS_CODEPAGE_865) [N/y/m/?] (NEW) Press Enter
Codepage 866 (Cyrillic/Russian) (CONFIG_NLS_CODEPAGE_866) [N/y/m/?] (NEW) Press Enter
Codepage 869 (Greek) (CONFIG_NLS_CODEPAGE_869) [N/y/m/?] (NEW) Press Enter
Simplified Chinese charset (CP936, GB2312) (CONFIG_NLS_CODEPAGE_936) [N/y/m/?] (NEW)
Press Enter
Traditional Chinese charset (Big5) (CONFIG_NLS_CODEPAGE_950) [N/y/m/?] (NEW) Press Enter
Japanese charsets (Shift-JIS, EUC-JP) (CONFIG_NLS_CODEPAGE_932) [N/y/m/?] (NEW) Press
Enter
Korean charset (CP949, EUC-KR) (CONFIG_NLS_CODEPAGE_949) [N/y/m/?] (NEW) Press Enter
Thai charset (CP874, TIS-620) (CONFIG_NLS_CODEPAGE_874) [N/y/m/?] (NEW) Press Enter
Hebrew charsets (ISO-8859-8, CP1255) (CONFIG_NLS_ISO8859_8) [N/y/m/?] (NEW) Press Enter
Windows CP1250 (Slavic/Central European Languages) (CONFIG_NLS_CODEPAGE_1250) [N/y/m/?]
(NEW) Press Enter
Windows CP1251 (Bulgarian, Belarusian) (CONFIG_NLS_CODEPAGE_1251) [N/y/m/?] (NEW) Press
Enter
NLS ISO 8859-1  (Latin 1; Western European Languages) (CONFIG_NLS_ISO8859_1) [N/y/m/?]
(NEW) Press Enter
NLS ISO 8859-2  (Latin 2; Slavic/Central European Languages) (CONFIG_NLS_ISO8859_2)
[N/y/m/?] (NEW) Press Enter
NLS ISO 8859-3  (Latin 3; Esperanto, Galician, Maltese, Turkish) (CONFIG_NLS_ISO8859_3)
[N/y/m/?] (NEW) Press Enter
NLS ISO 8859-4  (Latin 4; old Baltic charset) (CONFIG_NLS_ISO8859_4) [N/y/m/?] (NEW)
Press Enter
NLS ISO 8859-5  (Cyrillic) (CONFIG_NLS_ISO8859_5) [N/y/m/?] (NEW) Press Enter
NLS ISO 8859-6  (Arabic) (CONFIG_NLS_ISO8859_6) [N/y/m/?] (NEW) Press Enter
NLS ISO 8859-7  (Modern Greek) (CONFIG_NLS_ISO8859_7) [N/y/m/?] (NEW) Press Enter
NLS ISO 8859-9  (Latin 5; Turkish) (CONFIG_NLS_ISO8859_9) [N/y/m/?] (NEW) Press Enter
NLS ISO 8859-13 (Latin 7; Baltic) (CONFIG_NLS_ISO8859_13) [N/y/m/?] (NEW) Press Enter
NLS ISO 8859-14 (Latin 8; Celtic) (CONFIG_NLS_ISO8859_14) [N/y/m/?] (NEW) Press Enter
NLS ISO 8859-15 (Latin 9; Western European Languages with Euro) (CONFIG_NLS_ISO8859_15)
[N/y/m/?] (NEW) Press Enter
NLS KOI8-R (Russian) (CONFIG_NLS_KOI8_R) [N/y/m/?] (NEW) Press Enter
NLS KOI8-U/RU (Ukrainian, Belarusian) (CONFIG_NLS_KOI8_U) [N/y/m/?] (NEW) Press Enter
NLS UTF8 (CONFIG_NLS_UTF8) [N/y/m/?] (NEW) Press Enter
*
* Console drivers
*
VGA text console (CONFIG_VGA_CONSOLE) [Y/n/?] Press Enter
Video mode selection support (CONFIG_VIDEO_SELECT) [N/y/?] Press Enter
*
* Sound
*
Sound card support (CONFIG_SOUND) [Y/m/n/?] n
*
* USB support
*
Support for USB (CONFIG_USB) [Y/m/n/?] n
*
* USB Controllers
*
*
* USB Device Class drivers
*
*
*   SCSI support is needed for USB Storage
*
*
* USB Human Interface Devices (HID)
*
*
*   Input core support is needed for USB HID
*
*
* USB Imaging devices
*
```

```
*
* USB Multimedia devices
*
*
*   Video4Linux support is needed for USB Multimedia device support
*
*
* USB Network adaptors
*
*
* USB port drivers
*
*
* USB Serial Converter support
*
*
* USB Miscellaneous drivers
*
*
* Kernel hacking
*
Kernel debugging (CONFIG_DEBUG_KERNEL) [N/y/?] Press Enter

*
* Grsecurity
*
Grsecurity (CONFIG_GRKERNSEC) [N/y/?] y
Security level (Low, Medium, High, Customized) [Customized] Press Enter
*
* Buffer Overflow Protection
*
Openwall non-executable stack (CONFIG_GRKERNSEC_STACK) [N/y/?] y
Gcc trampoline support (CONFIG_GRKERNSEC_STACK_GCC) [N/y/?] Press Enter
Read-only kernel memory (CONFIG_GRKERNSEC_KMEM) [N/y/?] y
*
* Access Control Lists
*
Grsecurity ACL system (CONFIG_GRKERNSEC_ACL) [N/y/?] y
ACL Debugging Messages (CONFIG_GR_DEBUG) [N/y/?] y
Extra ACL Debugging Messages (CONFIG_GR_SUPERDEBUG) [N/y/?] Press Enter
Denied capability logging (CONFIG_GRKERNSEC_ACL_CAPLOG) [N/y/?] y
Path to gradm (CONFIG_GRADM_PATH) [/sbin/gradm] Press Enter
Maximum tries before password lockout (CONFIG_GR_MAXTRIES) [3] 2
Time to wait after max password tries, in seconds (CONFIG_GR_TIMEOUT) [30] Press Enter
*
* Filesystem Protections
*
Proc restrictions (CONFIG_GRKERNSEC_PROC) [N/y/?] y
Restrict to user only (CONFIG_GRKERNSEC_PROC_USER) [N/y/?] y
Additional restrictions (CONFIG_GRKERNSEC_PROC_ADD) [N/y/?] y
Linking restrictions (CONFIG_GRKERNSEC_LINK) [N/y/?] y
FIFO restrictions (CONFIG_GRKERNSEC_FIFO) [N/y/?] y
Secure file descriptors (CONFIG_GRKERNSEC_FD) [N/y/?] y
Chroot jail restrictions (CONFIG_GRKERNSEC_CHROOT) [N/y/?] y
Restricted signals (CONFIG_GRKERNSEC_CHROOT_SIG) [N/y/?] y
Deny mounts (CONFIG_GRKERNSEC_CHROOT_MOUNT) [N/y/?] y
Deny double-chroots (CONFIG_GRKERNSEC_CHROOT_DOUBLE) [N/y/?] y
Enforce chdir("/") on all chroots (CONFIG_GRKERNSEC_CHROOT_CHDIR) [N/y/?] y
Deny (f)chmod +s (CONFIG_GRKERNSEC_CHROOT_CHMOD) [N/y/?] y
Deny mknod (CONFIG_GRKERNSEC_CHROOT_MKNOD) [N/y/?] y
Deny ptraces (CONFIG_GRKERNSEC_CHROOT_PTRACE) [N/y/?] y
Restrict priority changes (CONFIG_GRKERNSEC_CHROOT_NICE) [N/y/?] y
Capability restrictions within chroot (CONFIG_GRKERNSEC_CHROOT_CAPS) [N/y/?] Press Enter
Secure keymap loading (CONFIG_GRKERNSEC_KBMAP) [N/y/?] y
*
* Kernel Auditing
*
Single group for auditing (CONFIG_GRKERNSEC_AUDIT_GROUP) [N/y/?] Press Enter
Exec logging (CONFIG_GRKERNSEC_EXECLOG) [N/y/?] Press Enter
Log execs within chroot (CONFIG_GRKERNSEC_CHROOT_EXECLOG) [N/y/?] y
Chdir logging (CONFIG_GRKERNSEC_AUDIT_CHDIR) [N/y/?] Press Enter
```

```
(Un)Mount logging (CONFIG_GRKERNSEC_AUDIT_MOUNT) [N/y/?] Press Enter
IPC logging (CONFIG_GRKERNSEC_AUDIT_IPC) [N/y/?] y
Ptrace logging (CONFIG_GRKERNSEC_AUDIT_PTRACE) [N/y/?] Press Enter
Signal logging (CONFIG_GRKERNSEC_SIGNAL) [N/y/?] y
Fork failure logging (CONFIG_GRKERNSEC_FORKFAIL) [N/y/?] y
Set*id logging (CONFIG_GRKERNSEC_SUID) [N/y/?] Press Enter
Log set*ids to root (CONFIG_GRKERNSEC_SUID_ROOT) [N/y/?] y
Time change logging (CONFIG_GRKERNSEC_TIME) [N/y/?] y
*
* Executable Protections
*
Exec process limiting (CONFIG_GRKERNSEC_EXECVE) [N/y/?] y
Dmesg(8) restriction (CONFIG_GRKERNSEC_DMESG) [N/y/?] y
Randomized PIDs (CONFIG_GRKERNSEC_RANDPID) [N/y/?] y
Altered default IPC permissions (CONFIG_GRKERNSEC_IPC) [N/y/?] Press Enter
Limit uid/gid changes to root (CONFIG_GRKERNSEC_TTYROOT) [N/y/?] y
Deny physical consoles (tty) (CONFIG_GRKERNSEC_TTYROOT_PHYS) [N/y/?] Press Enter
Deny serial consoles (ttyS) (CONFIG_GRKERNSEC_TTYROOT_SERIAL) [N/y/?] y
Deny pseudo consoles (pty) (CONFIG_GRKERNSEC_TTYROOT_PSEUDO) [N/y/?] Press Enter
Fork-bomb protection (CONFIG_GRKERNSEC_FORKBOMB) [N/y/?] y
GID for restricted users (CONFIG_GRKERNSEC_FORKBOMB_GID) [1006] Press Enter
Forks allowed per second (CONFIG_GRKERNSEC_FORKBOMB_SEC) [40] Press Enter
Maximum processes allowed (CONFIG_GRKERNSEC_FORKBOMB_MAX) [20] 35
Trusted path execution (CONFIG_GRKERNSEC_TPE) [N/y/?] y
Glibc protection (CONFIG_GRKERNSEC_TPE_GLIBC) [N/y/?] y
Partially restrict non-root users (CONFIG_GRKERNSEC_TPE_ALL) [N/y/?] y
GID for untrusted users: (CONFIG_GRKERNSEC_TPE_GID) [1005] Press Enter
Restricted ptrace (CONFIG_GRKERNSEC_PTRACE) [N/y/?] y
Allow ptrace for group (CONFIG_GRKERNSEC_PTRACE_GROUP) [N/y/?] Press Enter
*
* Network Protections
*
Randomized IP IDs (CONFIG_GRKERNSEC_RANDID) [N/y/?] y
Randomized TCP source ports (CONFIG_GRKERNSEC_RANDSRC) [N/y/?] y
Randomized RPC XIDs (CONFIG_GRKERNSEC_RANDRPC) [N/y/?] y
Altered Ping IDs (CONFIG_GRKERNSEC_RANDPING) [N/y/?] y
Randomized TTL (CONFIG_GRKERNSEC_RANDTTL) [N/y/?] y
Socket restrictions (CONFIG_GRKERNSEC_SOCKET) [N/y/?] y
Deny any sockets to group (CONFIG_GRKERNSEC_SOCKET_ALL) [N/y/?] y
GID to deny all sockets for: (CONFIG_GRKERNSEC_SOCKET_ALL_GID) [1004] Press Enter
Deny client sockets to group (CONFIG_GRKERNSEC_SOCKET_CLIENT) [N/y/?] Press Enter
Deny server sockets to group (CONFIG_GRKERNSEC_SOCKET_SERVER) [N/y/?] Press Enter
*
* Sysctl support
*
Sysctl support (CONFIG_GRKERNSEC_SYSCTL) [N/y/?] Press Enter
*
* Miscellaneous Features
*
Seconds in between log messages(minimum) (CONFIG_GRKERNSEC_FLOODTIME) [30] Press Enter
BSD-style coredumps (CONFIG_GRKERNSEC_COREDUMP) [N/y/?] y

*** End of Linux kernel configuration.
*** Check the top-level Makefile for additional configuration.
*** Next, you must run 'make dep'.
```

Compiling the Kernel

This section applies to both **Monolithic** and **Modularized** kernels. Once the kernel configuration has been completed, return to the `/usr/src/linux` directory (if you are not already in it) and compile the new kernel. You do so by using the following command:

- To compile the Kernel, use the following command:
 `[root@deep linux]#` **`make dep; make clean; make bzImage`**

This line contains three commands in one. The first one, **`make dep`**, actually takes your configuration and builds the corresponding dependency tree. This process determines what gets compiled and what doesn't. The next step, **`make clean`**, erases all previous traces of a compilation so as to avoid any mistakes in which the wrong version of a feature gets tied into the kernel. Finally, **`make bzImage`** does the full compilation of the kernel.

After the process is complete, the kernel is compressed and ready to be installed on your system. Before we can install the new kernel, we must know if we need to compile the corresponding modules. This is required ONLY if you said **yes** to "**`Enable loadable module support (CONFIG_MODULES)`**" and have compiled some options in the kernel configuration above as a module (See `Modularized kernel` configuration). In this case, you must execute the following commands:

- To compile the corresponding modules for your kernel, use the following commands:
 `[root@deep linux]#` **`make modules`**
 `[root@deep linux]#` **`make modules_install`**

> **WARNING:** The **`make modules`** and **`make modules_install`** commands are required ONLY if you say **yes** to "**`Enable loadable module support (CONFIG_MODULES)`**" in your kernel configurations (See `Modularized kernel` configuration) because you want to build a **`modularized kernel`**.

Installing the Kernel

This section applies to both the **Monolithic** and **Modularized** kernel. Ok, the kernel has been configured, compiled and is now ready to be installed your system. Below are the steps required to install all the necessary kernel components in your system.

Step 1
Copy the file `/usr/src/linux/arch/i386/boot/bzImage` from the kernel source tree to the `/boot` directory, and give it an appropriate new name.

- To copy the **`bzImage`** file to the `/boot` directory, use the following commands:
 `[root@deep /]#` **`cd /usr/src/linux/`** (if you are not already in it)
 `[root@deep linux]#` **`cp arch/i386/boot/bzImage /boot/vmlinuz-2.4.18`**

Step 2
A new System.map file is generated when you compile a kernel, and is a list of all the addresses in that kernel and their corresponding symbols. Every time that you create a new kernel, such a file System.map is created and saved in /usr/src/linux. It's a text file, which is read by a few programs to do address <-> symbol translation, and which you need if you ever get an Oops.

Certain commands, like klog, ps, and lsof, use the System.map file to get the name of kernel symbols. Without it some commands like lsof will complain that they can't find a System.map file to match the currently booted kernel.

Copy the file /usr/src/linux/System.map from the kernel source tree to the /boot directory, and give it an appropriate new name.

- To copy the **System.map** file to the /boot directory, use the following commands:
 [root@deep /]# **cd /usr/src/linux/** (if you are not already in it)
 [root@deep linux]# **cp System.map /boot/System.map-2.4.18**

Step 3
Move into the /boot directory and rebuild the links **vmlinuz** and **System.map**.

- To rebuild the **vmlinuz** and **System.map** files, use the following commands:
 [root@deep linux]# **cd /boot/**
 [root@deep /boot]# **ln -fs vmlinuz-2.4.18 vmlinuz**
 [root@deep /boot]# **ln -fs System.map-2.4.18 System.map**

We must rebuild the links of **vmlinuz** and **System.map** to point them to the new installed kernel version. Without the new links LILO or GRUB program will look, by default, for the old version of your Linux kernel.

Step 4
Remove obsolete and unnecessary files under the /boot directory to increase disk space:

- To remove obsolete and unnecessary files under the /boot directory, use commands:
 [root@deep /]# **cd /boot/** (if you are not already in it)
 [root@deep /boot]# **rm -f module-info**
 [root@deep /boot]# **rm -f initrd-2.4.x.img**

The **module-info** is a link, which points to the old modules directory of your original kernel. Since we have installed a brand new kernel, we don't need to keep this broken link.

The **initrd-2.4.x.img** is a file that contains an initial RAM disk image that serves as a system before the disk is available. This file is only available and is installed by the Linux initial setup installation if your system has a SCSI adapter present and only if your system has a SCSI adapter. If we use and have a SCSI system, the required driver now will have been incorporated into our new Linux kernel since we have build it by answering Y to the question related to our SCSI model during the configuration of the kernel, so we can safely remove this file (initrd-2.4.x.img).

Step 5
Create a new Linux kernel directory that will handle all the header files related to Linux kernel for future compilation of other programs on your system.

Recall, we had created two symlinks under the `/usr/include` directory that pointed to the Linux kernel header files to be able to compile it without receiving errors and also be able to compile future programs. The `/usr/include` directory is where all the header files for your Linux system are kept for reference and dependencies when you compile and install new programs.

The `asm`, and `linux` links are used when programs need to know some functions which are compile-time specific to the kernel installed on your system. Programs call other headers as well in the `/usr/include` directory when they must know specific information, dependencies, etc of your system.

- To create a new Linux kernel directory to handle all header files, use the commands:

```
[root@deep /]# mkdir -p /usr/src/linux-2.4.18/include
[root@deep /]# cd /usr/src/linux/
[root@deep linux]# cp -r include/asm-i386 ../linux-2.4.18/include/
[root@deep linux]# cp -r include/linux ../linux-2.4.18/include/
[root@deep linux]# cd ../
[root@deep src]# rm -rf /usr/src/linux
[root@deep src]# cd /usr/src/ (to be sure that we are into the src directory)
[root@deep src]# ln -s /usr/src/linux-2.4.18 linux
```

First we create a new directory named "`linux-2.4.18`" based on the version of the kernel we have installed for easy interpretation, then we copy directories `asm-i386`, and `linux` from `/usr/src/linux/include` to our new location `/usr/src/linux-2.4.18/include`. After we remove the entire source directory where we compiled the new kernel, we create a new symbolic link named "`linux`" under `/usr/src` that points to our new `/usr/src/linux-2.4.18` directory. With these steps, future compiled programs will know where to look for headers related to the kernel on your server.

NOTE: This step will allow us to gain space on our hard drive and will reduce the security risks. The Linux kernel source directory handles a lot files and is about **94M** in size when uncompressed. With the procedure described above, our Linux kernel directory began approximately **4M** in size so we save **90MB** for the same functionalities.

Verifying or upgrading your boot loader

Once the new kernel image has been installed on your server, we have to inform our boot loader about it. This is done inside the configuration file of your boot loader software. Below I show you how to do it depending if you use `GRUB` or `LILO` as your boot loader.

LILO:
This step applies only if you use `LILO` as your boot loader on the system. If you use `GRUB` as your boot loader instead of `LILO` (highly recommended), then you can skip this section and go directly to the next one.

Step 1
You need to edit the `lilo.conf` file to make your new kernel one of the boot time options:

- Edit the **`lilo.conf`** file (`vi /etc/lilo.conf`) and make the appropriate change on the line that reads "`image=/boot/vmlinuz-x.x.x`".

```
[root@deep /]# vi /etc/lilo.conf

boot=/dev/sda
map=/boot/map
install=/boot/boot.b
timeout=00
default=linux
restricted
password=somepasswd

image=/boot/vmlinuz
label=linux
read-only
root=/dev/sda6
```

Step 2
Once the necessary modifications have been made in the `/etc/lilo.conf` file as shown above, we update our `lilo.conf` file for the change to take effect.

- This can be done with the following command:

```
[root@deep /]# /sbin/lilo
Added linux *
```

GRUB:
This step applies only if you use `GRUB` as your boot loader on the system. In most case, `GRUB` does not need to be updated when you install a new kernel on your computer but here we have to verify inside our `grub.conf` file if all default setting are still available and configured because when we uninstall Red Hat kernel `RPM` package, the software automatically remove some needed parameters from the `GRUB` configuration file.

Step 1
Edit your `GRUB` configuration file and be sure that everything is correct and look like the following. Your setting should differ from the example below.

- Edit the **`grub.conf`** file (`vi /etc/grub.conf`) and check your setting.

```
[root@deep /]# vi /etc/grub.conf

default 0
timeout 00
title  Red Hat Linux
kernel (hd0,0)/vmlinuz ro root=/dev/sda5
```

Reconfiguring `/etc/modules.conf` file

This section applies only if you chose to install a **Modularized** Kernel on your system. The `/etc/modules.conf` file represents the (optional) configuration file for loading kernel modules on your system. It is used to modify the behavior of `modprobe` and `depmod` programs.

This file consists of a set of lines with different parameters. It is important after each upgrade of a `modularized kernel` to verify if all the information and parameters contained inside it are valid and correct.

All the contents of the `/etc/modules.conf` file apply only for systems where the kernel has been configured with modules (`modularized kernel`). So if you have recompiled your new kernel with some new options as modules or if you have removed some modules from it, it is important to update or remove the **`modules.conf`** file to reflect the changes and eliminate possible error message during booting.

As an example, the following is the content of the **`modules.conf`** file on my system. Linux has added these parameters automatically, depending of the system hardware during the primary install stage of the operating system.

```
alias scsi_hostadapter aic7xxx
alias eth0 eepro100
alias eth1 eepro100
alias parport_lowlevel parport_pc
alias usb-controller uhci
```

One important use of the `modules.conf` file is the possibility of using the "`alias`" directive to give alias names to modules and link object files to a module.

After recompilation of the kernel, and depending on how we have answered the different kernel questions during kernel configuration, it may be possible that we need to make some adjustments to the default parameters, especially if we have answered **`yes`** during kernel configuration to some devices available in our system, like network cards and `SCSI` adapters.

If the configuration file `/etc/modules.conf` is missing, or if any directive is not overridden, the default will be to look under `/lib/modules` directory containing modules compiled for the current release of the kernel. Therefore, we can remove the `/etc/modules.conf` file from the system and let the `modprobe` and `depmod` programs manage all existing modules for us.

To summarize, you can:

1) Keep the **`modules.conf`** file; only kernel options which you have answered **m** during kernel configuration time (of course only if these modules did exist into `modules.conf`). Any kernel options where you have answered **yes** or **no** will not appear into the **`modules.conf`** file.

2) Or remove the `/etc/modules.conf` file from your system and let `modprobe` and `depmod` programs manage all existing modules for you. On a server environment, I prefer to use this choice.

Rebooting your system to load the new kernel

Whether you have installed a new **Monolithic** Kernel where codes and drivers are compiled into the kernel and are always loaded or a **Modularized** Kernel where some segment of codes are compiled into the kernel as a module and loaded on demand, it is time to **Reboot** your system and test your results.

- To reboot your Linux system, use the following command:
  ```
  [root@deep /]# reboot
  ```

When the system is rebooted and you are logged in, verify the new version of your kernel with the following command:

- To verify the version of your new kernel, use the following command:
  ```
  [root@deep /]# uname -a
  Linux dev 2.4.18-grsec-1.9.4 #1 Wed Jun 19 15:14:55 EDT 2002 i686 unknown
  ```

Congratulations!

Delete programs, edit files pertaining to modules

This section applies only if you chose to install a **Monolithic** Kernel on your system. By default when you install Linux for the first time (like we did), the kernel is built as a Modularized Kernel. This means that each device or function we need exists as a module and is controlled by the Kernel Daemon program named **`kmod`**. `kmod` automatically loads some modules and functions into memory as they are needed, and unloads them when they're no longer being used.

`kmod` and other module management programs included in the **`modutils`** `RPM` package use the `modules.conf` file located in the `/etc` directory to know for example which Ethernet card you have, if your Ethernet card requires special configuration and so on. If we don't use any modules in our newly compiled kernel because we have compiled the kernel as a Monolithic Kernel and ONLY in this case, we can remove the `modules.conf` file and completely uninstall the **`modutils`** `RPM` package.

- To remove the **`modules.conf`** file, use the following command:
  ```
  [root@deep /]# rm -f /etc/modules.conf
  ```

- To uninstall the **`modutils`** package, use the following command:
  ```
  [root@deep /]# rpm -e --nodeps modutils
  ```

> **WARNING:** Once again, the above is required only if you said **no** to "`Enable loadable module support (CONFIG_MODULES)`" in your kernel configuration because you have decided to build a Monolithic Kernel.

Making a new rescue floppy for Modularized Kernel

This section applies only if you chose to install a **Modularized** Kernel on your system. Immediately after you successfully start your system and log in as root, you should create a new emergency boot floppy disk. The procedure to achieve it is the same as shown at the beginning of this chapter related to Linux Kernel.

Please go back to the beginning of this chapter and follow the procedures to recreate a new emergency boot floppy disk suitable for the new install Linux kernel on your system. Don't forget to test the boot disk to be sure that it works.

The `mkbootdisk` program runs only on Modularized Kernel. So you can't use it on a Monolithic Kernel; instead create an emergency boot floppy disk for Monolithic kernel as shown below.

Making a emergency boot floppy disk for Monolithic Kernel

This section applies only if you chose to install a **Monolithic** Kernel in your system. Because it is possible to create a rescue floppy only on modularized kernel, we must find another way to boot our Linux system for a monolithic kernel if the Linux kernel on the hard disk is damaged.

This is possible with a Linux emergency boot floppy disk. You should create it immediately after you successfully start your system and log in as root. To create the emergency boot floppy, follow these steps:

Step 1
Insert a floppy disk and format it with the following command:

```
[root@deep /]# fdformat /dev/fd0H1440
Double-sided, 80 tracks, 18 sec/track. Total capacity 1440 kB.
Formatting ... done
Verifying ... done
```

Step 2
Copy the actual file "`vmlinuz`" from the `/boot` directory to the floppy disk:

```
[root@deep /]# cp /boot/vmlinuz /dev/fd0H1440
cp: overwrite `/dev/fd0H1440'? y
```

NOTE: The `vmlinuz` file is a symbolic link that points to the real Linux kernel.

Step 3
Determine the kernel's root device with the following command:

```
[root@deep /]# rdev
/dev/sda6 /
```

NOTE: The kernel's root device is the disk partition where the root file system is located. In this example, the root device is `/dev/sda6`; the device name should be different on your system.

Step 4
Set the kernel's root device with the following command:
```
[root@deep /]# rdev /dev/fd0H1440 /dev/sda6
```

NOTE: To set the kernel's root device, use the device reported by the "`rdev`" command utility in the previous step.

Step 5
Mark the root device as read-only with the following command:
```
[root@deep /]# rdev -R /dev/fd0H1440 1
```

NOTE: This causes Linux to initially mount the root file system as read-only. By setting the root device as read-only, you avoid several warnings and error messages.

Step 6
Now put the boot floppy in the drive A: and reboot your system with the following command to be sure that your new boot disk is working:
```
[root@something /]# reboot
```

Following these guidelines, you will now have a boot floppy with a known working kernel in case of problems with the upgrade. I recommend rebooting the system with the floppy to make sure that the floppy works correctly.

Step 7
Because the `mkbootdisk` and `dosfstools` program are required only when you have a Modularized kernel installed in your Linux system, we can remove the unneeded `mkbootdisk` and `dosfstools` packages from the system when we have a Monolithic kernel installed on our server.

- To uninstall the `mkbootdisk` and `dosfstools` utility, use the following command:
  ```
  [root@deep /]# rpm –e mkbootdisk dosfstools
  ```

CHAPTER 7

Process file system management

IN THIS CHAPTER

1. **What is `sysctl`?**
2. **`/proc/sys/vm`: The virtual memory subsystem of Linux**
3. **`/proc/sys/fs`: The file system data of Linux**
4. **`/proc/sys/net/ipv4`: IPV4 settings of Linux**
5. **Other possible optimization of the system**

Linux /proc

Abstract

The /proc (the process file system), also known as a pseudo-filesystem, is used as an interface to kernel data structures. It doesn't exist, neither the /proc directory nor its subdirectories or its files actually exist. Most of the files in this special directory are read-only and cannot be changed, but some kernel variables can be changed. It is these files that we will talk about in this chapter of the book.

It is important to note that the /proc filesystem is structured in a hierarchy. Most of the entries in the /proc directory are a decimal number, corresponding to a process-ID running on the system. These entries are themselves subdirectories and access to process state that is provided by additional files contained within each subdirectory. Have you ever thought about where all the processes running in the background of your system are handled and managed by the kernel? The answer is the /proc filesystem directory of Linux.

But the /proc filesystem doesn't handle only process ID of the system; it is also responsible for providing and managing all access to the state of each information on the system. This information is comprised of CPU, devices, IDE, SCSI, interrupts, io-ports, memories, modules, partitions, PCI information and much more. Just take a quick look inside your /proc filesystem directory to get an idea of the available features controlled by the kernel through the /proc filesystem. We can read the contents of this information to get an idea of what processor, PCI, network cards, kernel version, partitions, etc that we have on our system.

As we said before, not all features available in the /proc filesystem are customizable, most are managed by the kernel and cannot be changed. Most are well controlled by the kernel and should not require any modifications since the kernel does a good job with them. Some can, and need to be, changed and customized to better fit your system resources, and increase security. It is those customizable features related to performance and security of the Linux system under the /proc filesystem that we will explain and customize in this chapter.

This is possible with the /etc/sysctl.conf file which contains values that change the default parameters of customizable features in the /proc filesystem. To recap, systcl.conf is the configuration file that talks to sysctl(8) which is an interface that allows you to make changes to a running Linux system. We use systcl.conf to talk to the kernel and say for example: *hey, I need more power on the virtual memory, please change your value to this value*.

Throughout this chapter, we'll often use it to customize our /proc filesystem on Linux to better utilize resources, power and security of our particular machine. Remember that everyone have a different computer with different hardware, setting and this is why changing some default customizable values in the /proc directory could make the difference on security and speed.

In this chapter, we will talk about customized parameters available under the /proc/sys directory since most of all changeable parameters are located under this directory. We will talk about virtual memory, file system, TCP/IP stack security and performance.

What is `sysctl`?

`sysctl` is an interface that allows you to make changes to a running Linux system. It serves two functions: to read and to modify system settings.

- To view all readable variables, use the following command:
  ```
  [root@deep /]# sysctl -a
  ```

- To read a particular variable, for example, `fs.file-max`, use the following command:
  ```
  [root@deep /]# sysctl fs.file-max
  fs.file-max = 8192
  ```

- To set a particular variable for `fs.file-max`, use the following command:
  ```
  [root@deep /]# sysctl -w fs.file-max=5536
  fs.file-max = 16384
  ```

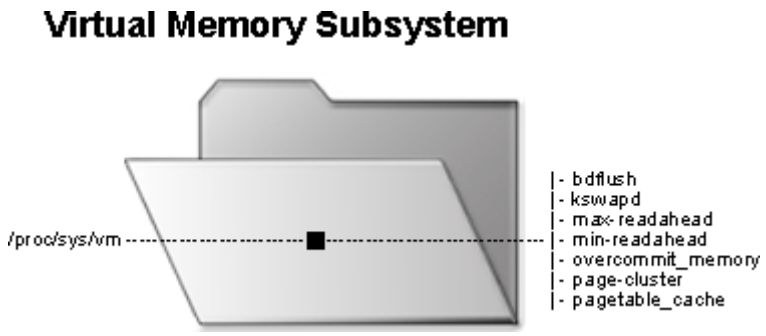
Settings of `sysctl` variables are usually strings, numbers, or booleans (a boolean being 1 for yes or a 0 for no). If you set and change variable manually with the `sysctl` command as show above, your changes will not resists on the next reboot of the system. For this reason, we will use and show you further down in this chapter how to make your changes permanent even on possible reboot of the server by using the `/etc/sysctl.conf` file.

`/proc/sys/vm`: The virtual memory subsystem of Linux

All parameters described in this chapter reside under the `/proc/sys/vm` directory of the server and can be used to tune the operation of the **v**irtual **m**emory (`VM`) subsystem of the Linux kernel. Be very careful when attempting this. You can optimize your system, but you can also cause it to crash. Since every system is different, you'll probably want some control over this piece of the system.

Finally, these are advanced setting and if you don't understand them, then don't try to play in this area or try to use all the examples below in your system. Remember that all systems are different and require different settings and customizations. The majority of the following hacks will work fine on a server with >= at 512MB of `RAM` or a minimum of 256MB of `RAM`. Below this amount of memory, nothing is guaranteed and the default setting will just be fine for you.

Next I'll show you parameters that can be optimized. All suggestions I make in this section are valid for all kinds of servers. The only difference depends on the amount of `RAM` your machine has and this is where the settings will change.



The above figure shows a snapshot of `/proc/sys/vm` directory on an OpenNA Linux & Red Hat Linux system running kernel version 2.4. Please note that this picture may look different on your system.

The `bdflush` parameters:

The `bdflush` file is closely related to the operation of the virtual memory (VM) subsystem of the Linux kernel and has a little influence on disk usage. This file `/proc/sys/vm/bdflush` controls the operation of the `bdflush` kernel daemon. We generally tune this file to improve file system performance.

By changing some values from the defaults shown below, the system seems more responsive; e.g. it waits a little more to write to disk and thus avoids some disk access contention. The `bdflush` parameters currently contain 9 integer values, of which 4 are actually used by the kernel. Only first, fifth, sixth and the seventh parameters are used by the kernel for `bdflush` setup and all the other parameters are not used and their values are set to '`0`'.

Parameter 1 (`nfract`):
The `bdflush` parameter 1 governs the maximum number of dirty buffers in the buffer cache. Dirty means that the contents of the buffer still have to be written to disk (as opposed to a clean buffer, which can just be forgotten about). Setting this to a high value means that Linux can delay disk writes for a long time, but it also means that it will have to do a lot of `I/O` (Input/Output) at once when memory becomes short. A low value will spread out disk `I/O` more evenly at the cost of more frequent `I/O` operations. The default value is 40%, the minimum is 0%, and the maximum is 100%. We improve the default value here.

Parameter 2 (`dummy1`):
This parameter is unused by the system so we don't need to change the default ones.

Parameter 3 (`dummy2`):
This parameter is unused by the system so we don't need to change the default ones.

Parameter 4 (`dummy3`):
This parameter is unused by the system so we don't need to change the default ones.

Parameter 5 (`interval`):
The `bdflush` parameter 5 specifies the minimum rate at which `kupdate` will wake and flush. The value is expressed in jiffies (clockticks), the number of jiffies per second is normally 100. Thus, x*HZ is x seconds. The default value is 5 seconds, the minimum is 0 seconds, and the maximum is 600 seconds. We keep the default value here.

Parameter 6 (`age_buffer`):
The `bdflush` parameter 6 governs the maximum time Linux waits before writing out a dirty buffer to disk. The value is in jiffies. The default value is 30 seconds, the minimum is 1 second, and the maximum 6,000 seconds. We keep the default value here.

Parameter 7 (`nfract_sync`):
The `bdflush` parameter 7 governs the percentage of buffer cache that is dirty before `bdflush` activates synchronously. This can be viewed as the hard limit before `bdflush` forces buffers to disk. The default is 60%, the minimum is 0%, and the maximum is 100%. We improve the default value here.

Parameter 8 (`dummy4`):
This parameter is unused by the system so we don't need to change the default ones.

Parameter 9 (`dummy5`):
This parameter is unused by the system so we don't need to change the default ones.

The default kernel setup for the `bdflush` parameters is:
"40 64 64 256 500 3000 60 0 0"

The default setup for the `bdflush` parameters under OpenNA Linux is:
"60 64 64 256 500 3000 80 0 0"

The default setup for the `bdflush` parameters under Red Hat Linux is:
"30 64 64 256 500 3000 60 0 0"

Step 1
To change the values of **`bdflush`**, type the following command on your terminal:

- Edit the **`sysctl.conf`** file (vi `/etc/sysctl.conf`) and add the following line:

```
# Improve file system performance
vm.bdflush = 60 64 64 256 500 3000 80 0 0
```

Step 2
You must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

NOTE: There is another way to update the entry without restarting the network by using the following command in your terminal screen:
`[root@deep /]# sysctl -w vm.bdflush="60 64 64 256 500 3000 80 0 0"`

The `kswapd` parameter:

The `kswapd` file is related to the kernel swapout daemon. This file `/proc/sys/vm/kswapd` frees memory on the system when it gets fragmented or full. Its task is to keep the memory management system operating efficiently. Since every system is different, you'll probably want some control over this piece of the system.

There are three parameters to tune in this file and two of them (`tries_base` and `swap_cluster`) have the largest influence on system performance. The `kswapd` file can be used to tune the operation of the virtual memory (VM) subsystem of the Linux kernel.

Parameter 1 (`tries_base`):
The `kswapd` parameter 1 specifies the maximum number of pages `kswapd` tries to free in one round. Usually this number will be divided by 4 or 8, so it isn't as big as it looks. Increase this number to cause swap to be released faster, and increase overall swap throughput. The default value is 512 pages. We keep the default value here.

Parameter 2 (`tries_min`):
The `kswapd` parameter 2 specifies the minimum number of pages `kswapd` tries to free a least each time it is called. Basically it's just there to make sure that `kswapd` frees some pages even when it's being called with minimum priority. The default value is 32 pages. We keep the default value here.

Parameter 3 (`swap_cluster`):
The `kswapd` parameter 3 specifies the number of pages `kswapd` writes in one iteration. You want this large to increase performance so that `kswapd` does its I/O in large chunks and the disk doesn't have to seek often, but you don't want it to be too large since that would flood the request queue. The default value is 8 pages. We improve the default value here.

The default kernel setup for the `kswapd` parameters is:
"512 32 8"

The default setup for the `kswapd` parameters under OpenNA Linux is:
"512 32 32"

The default setup for the `kswapd` parameters under Red Hat Linux is:
"512 32 8"

Step 1
To change the values of **kswapd**, type the following command on your terminal:

- Edit the **sysctl.conf** file (`vi /etc/sysctl.conf`) and add the following lines:

```
# Increase swap bandwidth system performance
vm.kswapd = 512 32 32
```

Step 2
You must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

NOTE: There is another way to update the entry without restarting the network by using the following command into your terminal screen:
`[root@deep /]# sysctl -w vm.kswapd=”512 32 32”`

The `overcommit_memory` parameter:

The `overcommit_memory` parameter is simply a flag that enables memory overcommitment. Memory overcommitment is a procedure to check that a process has enough memory to allocate a new virtual mapping. When this flag is 0, the kernel checks before each `malloc()` to see if there's enough memory left. If the flag is 1, the system pretends there's always enough memory and don't make the check on the system. This feature can be very useful ONLY on big servers with a lot of pysical memories available (>= 2GB) because there are a lot of programs that `malloc()` huge amounts of memory "just-in-case" and don't use much of it.

The default kernel setup for the `overcommit_memory` parameter is:
"0"

The default setup for the `overcommit_memory` parameter under OpenNA Linux is:
"0"

The default setup for the `overcommit_memory` parameter under Red Hat Linux is:
"0"

Step 1
To change the value of **`overcommit_memory`**, type the following command on your terminal:

- Edit the **`sysctl.conf`** file (`vi /etc/sysctl.conf`) and add the following lines:

```
# Enables/Disables memory overcommitment
vm.overcommit_memory = 0
```

Step 2
You must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

WARNING: Only change the default value of 0 to become 1 on systems with more than 2GB of RAM. Recall that on small systems the value must be set to 0 (`overcommit_memory=0`).

There is another way to update the entry without restarting the network by using the following command into your terminal screen:

```
[root@deep /]# sysctl -w overcommit_memory=0
```

The `page-cluster` parameter:

The Linux virtual memory subsystem avoids excessive disk seeks by reading multiple pages on a page fault. The number of pages it reads is highly dependent on the amount of memory in your machine. The number of pages the kernel reads in at once is equal to 2 ^ page-cluster. Values above 2 ^ 5 don't make much sense for swap because we only cluster swap data in 32-page groups. The `page-cluster` parameter is used to tune the operation of the virtual memory (VM) subsystem of the Linux kernel.

The default kernel setup for the `kswapd` parameter is:
"3"

The default setup for the `kswapd` parameter under OpenNA Linux is:
"5"

The default setup for the `kswapd` parameter under Red Hat Linux is:
"4"

Step 1
To change the value of **`page-cluster`**, type the following command on your terminal:

- Edit the **`sysctl.conf`** file (`vi /etc/sysctl.conf`) and add the following lines:

```
# Increase number of pages kernel reads in at once
vm.page-cluster = 5
```

Step 2
You must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

NOTE: There is another way to update the entry without restarting the network by using the following command into your terminal screen:

```
[root@deep /]# sysctl -w vm.page-cluster=5
```

The `pagetable_cache` parameter:

The kernel keeps a number of page tables in a per-processor cache (this helps a lot on `SMP` systems). The cache size for each processor will be between the low and the high value. On `SMP` systems it is used so that the system can do fast pagetable allocations without having to acquire the kernel memory lock.

For large systems, the settings are probably OK. For normal systems they won't hurt a bit. For small systems (<16MB RAM) and on a low-memory, single CPU system it might be advantageous to set both values to 0 so you don't waste the memory.

The default kernel setup for the `kswapd` parameters is:
"25 50"

The default setup for the `kswapd` parameters under OpenNA Linux is:
"25 50"

The default setup for the `kswapd` parameters under Red Hat Linux is:
"25 50"

Step 1
To change the values of **`pagetable_cache`**, type the following command on your terminal:

- Edit the **`sysctl.conf`** file (`vi /etc/sysctl.conf`) and add the following lines:

```
# Improve number of page tables keeps in a per-processor cache
vm.pagetable_cache = 25 50
```

Step 2
You must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

WARNING: Only change these values on systems with multiple processors (`SMP`) or on small systems (single processor) with less than 16MB of RAM. Recall that on small systems the both values must be set to 0 (`vm.pagetable_cache = 0 0`).

There is another way to update the entry without restarting the network by using the following command into your terminal screen:

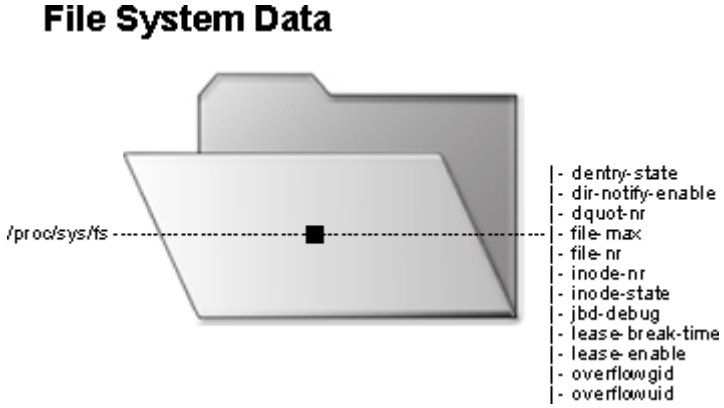
```
[root@deep /]# sysctl -w vm.pagetable_cache=”25 50”
```

`/proc/sys/fs`: The file system data of Linux

All parameters described later in this chapter reside under the `/proc/sys/fs` directory of the server and can be used to tune and monitor miscellaneous things in the operation of the Linux kernel. Be very careful when attempting this. You can optimize your system, but you can also cause it to crash. Since every system is different, you'll probably want some control over these pieces of the system.

Finally, these are advanced settings and if you don't understand them, then don't play in this area or try to use all the examples below in your system. Remember that all systems are different and required different setting and customization.

Below I show you only parameters that can be optimized for the system. All suggestions I enumerate in this section are valid for every kind of servers. The only difference depends on the amount of MB of RAM your machines have and this is where settings will change.



The above figure shows a snapshot of `/proc/sys/fs` directory on a OpenNA Linux & Red Hat Linux system running kernel version 2.4. Please note that this picture may look different on your system.

The `file-max` & `file-nr` parameters:

The `file-max` and `file-nr` files work together on Linux, we use the `file-max` parameter to sets the maximum number of file-handles that the Linux kernel will allocate and the `file-nr` file to get information about the number of allocated file handles, the number of used file handles and the maximum number of file handles presently on the system. A large-scale production server may easily require many thousands of file-handles, depending on the kind and number of services running concurrently on the server. On busy servers where many services are running concurrently, we generally tune this file (`file-max`) to improve the number of open files available on the system.

It is important to note that you need to increase the limit of open files available on your server ONLY when you get lots of error messages about running out of file handles. If you don't receive this kind of error message, you really DON'T need to increase the default value.

- To know the number of allocated file handles, the number of used file handles and the maximum number of file handles on your system, use the following command:

```
[root@deep /]# cat /proc/sys/fs/file-nr
405     137     8192
```

The first value (405) in our result is the number of allocated file handles, the second value (137) is the number of used file handles, and the last value (8192) is the maximum number of file handles. When the allocated file handles (405) come close to the maximum (8192), but the number of actually used ones (137) is far less, you've encountered a peak in your usage of file handles and you don't need to increase the maximum. The default kernel setup is suitable for most of us.

The default kernel setup for the `file-max` parameter is:
"8192"

The default setup for the `file-max` parameter under OpenNA Linux is:
"8192"

The default setup for the `file-max` parameter under Red Hat Linux is:
"8192"

Step 1
To adjust the value of **`file-max`**, type the following command on your terminal:

- Edit the **`sysctl.conf`** file (`vi /etc/sysctl.conf`) and add the following lines:

```
# Increase limit of file-handles
fs.file-max = 8192
```

Step 2
You must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

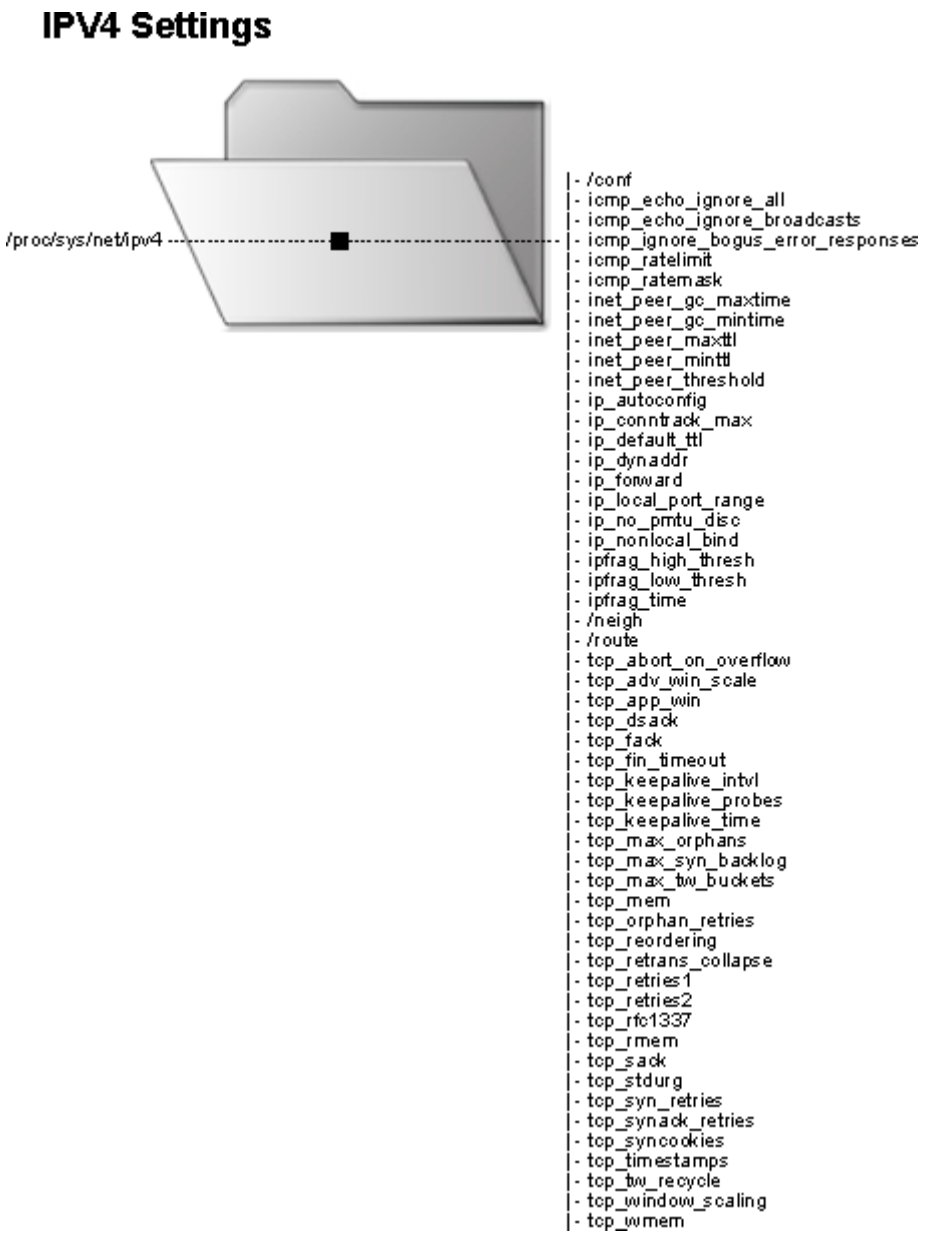
```
[root@deep /]# /etc/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

NOTE: There is another way to update the entry without restarting the network by using the following command into your terminal screen:

```
[root@deep /]# sysctl -w fs.file-max=8192
```

`/proc/sys/net/ipv4`: IPV4 settings of Linux

All parameters described below reside under the `/proc/sys/net/ipv4` directory of the server and can be used to control the behavior of the IPv4 subsystem of the Linux kernel. Below I show you only the parameters, which can be used for the network security of the system.



The above figure shows a snapshot of `/proc/sys/net/ipv4` directory on a OpenNA Linux & Red Hat Linux system running kernel version 2.4. Please note that this picture may look different on your system.

Prevent your system responding to `ping` request:

Preventing your system for responding to `ping` requests can make a big improvement in your network security since no one can `ping` your server and receive an answer. The `TCP/IP` protocol suite has a number of weaknesses that allows an attacker to leverage techniques in the form of covert channels to surreptitiously pass data in otherwise benign packets.

Step 1
Preventing your server from responding to `ping` requests can help to minimize this problem. Not responding to pings would at least keep most "crackers" out because they would never know it's there. `ICMP` blocking can hurt the performance of long-duration `TCP` connections, and this is due to the fact that `MTU` discovery relies on `ICMP` packets to work.

- Edit the **`sysctl.conf`** file (`vi /etc/sysctl.conf`) and add the following lines:

```
# Enable/Disable ignoring ping request
net.ipv4.icmp_echo_ignore_all = 1
```

When this key is on (`1`), the computer will ignore all `ICMP` packets. It is not recommended to turn on this key, except in special situations when you receive an `ICMP` packet based attack. In the above parameter, we enable this option.

Step 2
Once the configuration has been set, you must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

NOTE: There is another way to update the entry without restarting the network by using the following command into your terminal screen:

```
[root@deep /]# sysctl -w net.ipv4.icmp_echo_ignore_all=1
```

Refuse responding to `broadcasts` request:

As for the `ping` request, it's also important to disable `broadcast` requests. When a packet is sent to an IP `broadcast` address (i.e. `192.168.1.255`) from a machine on the local network, that packet is delivered to all machines on that network. Then all the machines on a network respond to this `ICMP` echo request and the result can be severe network congestion or outages (Denial-of-Service attacks). See the RFC 2644 for more information.

Step 1
To disable `broadcast` requests, type the following command on your terminal.

- Edit the **`sysctl.conf`** file (`vi /etc/sysctl.conf`) and add the following lines:

```
# Enable/Disable ignoring broadcasts request
net.ipv4.icmp_echo_ignore_broadcasts = 1
```

When this key is on (`1`), the server will never answer to "`ping`" if its destination address is `multicast` or `broadcast`. It is good to turn on (`1`) this key to avoid your server becoming an involuntary partner of a DoS attack. In the above parameter, we enable this option.

Step 2
Once the configuration has been set, you must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all networks devices manually on your system, use the following command:

```
[root@deep /]# /etc/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

NOTE: There is another way to update the entry without restarting the network by using the following command into your terminal screen:

```
[root@deep /]# sysctl -w net.ipv4.icmp_echo_ignore_broadcasts=1
```

Routing Protocols:

Routing and routing protocols can create several problems. IP source routing, where an IP packet contains details of the path to its intended destination, is dangerous because according to RFC 1122 the destination host must respond along the same path. If an attacker was able to send a source routed packet into your network, then he would be able to intercept the replies and fool your host into thinking it is communicating with a trusted host.

Step 1
I strongly recommend that you disable IP source routing on all network interfaces on the system to protect your server from this hole. If IP source routing is set to off (`0`), the server will not accept source-routed frames. Remember that Source-routed frames have an embedded, explicit description of the route between source and destination. Normally, IP packet routing is based solely on destination's address.

- Edit the **`sysctl.conf`** file (`vi /etc/sysctl.conf`) and add the following lines:

```
# Enable/Disable IP source routing
net.ipv4.conf.all.accept_source_route = 0
net.ipv4.conf.default.accept_source_route = 0
```

Source-routed packets are a powerful concept but were never used, and can bring security problems because they allow a non-blind, REMOTE spoof. It is a very good idea to turn off (`0`) these keys. In the above parameters, we disable these options.

Step 2
Once configurations have been set, you must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

NOTE: There is another way to update the entry without restarting the network by using the following command into your terminal screen:

```
[root@deep /]# sysctl -w net.ipv4.conf.all.accept_source_route=1
[root@deep /]# sysctl -w net.ipv4.conf.default.accept_source_route=1
```

Enable TCP SYN Cookie Protection:

A "SYN Attack" is a **D**enial **o**f **S**ervice (DoS) attack that consumes all the resources on your machine, forcing you to reboot. Denials of Service attacks (attacks which incapacitate a server due to high traffic volume or ones those tie-up system resources enough that the server cannot respond to a legitimate connection request from a remote system) are easily achievable from internal resources or external connections via extranets and Internet. Enabling `TCP SYN Cookie Protection` will help to eliminate the problem. Below is a simple explanation of the concept.

Every `TCP/IP` connection begins with a 3-way handshake:

1. Client sends a packet (packet 1) to server with `SYN` bit on, and waits;
2. Server sends a confirmation packet (packet 2) to client, and waits;
3. Client sends a third packet (packet 3) that consolidates the connection.

Once the 3-way handshake is done, the server keeps data of packet 1 in a queue to compare it with packet 3 and establish the connection. This queue is limited in size and a quite high timeout. The SYN-flood attack exploits this fact and sends a lot of type-1 packets with random IP source addresses; the phase 3 answers never arrive; and once the queue is full, the server cannot receive more connections, be they legitimate or forged.

The SYN cookie "trick" is to embed a code in the header of phase 2 packets, so server DOES NOT NEED TO KEEP any information about the client. If the phase 3 packet arrives someday, the server will calculate the port and client initial sequence number based solely on that packet - and will be able to establish the connection.

Step 1
Since this embedded codification reduces randomness of the server initial sequence number, and thus can increase the "chance" of IP spoof family attacks, `SYN` cookies are used only in emergency situations, that is, when the half-open connections queue is full.

- Edit the **`sysctl.conf`** file (`vi /etc/sysctl.conf`) and add the following lines:

```
# Enable/Disable TCP SYN Cookie Protection
net.ipv4.tcp_syncookies = 1
```

If this key is on (`1`), the kernel will send "SYN cookies" ONLY and ONLY when the half-open connections queue becomes full. This will mitigate the effects of SYN-flood DoS attacks. In the above parameter, we enable this option.

Step 2
Once the configuration has been set, you must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
```

WARNING: There is another way to update the entry without restarting the network by using the following command into your terminal screen:

```
[root@deep /]# sysctl -w net.ipv4.tcp_syncookies=1
```

Disable ICMP Redirect Acceptance:

When hosts use a non-optimal or defunct route to a particular destination, an ICMP redirect packet is used by routers to inform the hosts what the correct route should be. If an attacker is able to forge ICMP redirect packets, he or she can alter the routing tables on the host and possibly subvert the security of the host by causing traffic to flow via a path you didn't intend.

Step 1
A legitimate ICMP REDIRECT packet is a message from a router that says "router X is better than me to reach network Y". Therefore, in complex networks, it will be highly recommended to keep these keys activated. On simple networks, it's strongly recommended to disable ICMP Redirect Acceptance into all available interfaces on the server to protect it from this hole.

- Edit the **sysctl.conf** file (vi /etc/sysctl.conf) and add the following lines:

```
# Enable/Disable ICMP Redirect Acceptance
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.default.accept_redirects = 0
```

When these keys are off (0), the kernel does not honor ICMP_REDIRECT packets, thus avoiding a whole family of attacks based on forging of this type of packet. In the above parameters, we disable these options.

Step 2
Once the configurations have been set, you must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all networks devices manually on your system, use the following command:

```
[root@deep /]# /etc/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

NOTE: There is another way to update the entry without restarting the network by using the following command into your terminal screen:

```
[root@deep /]# sysctl -w net.ipv4.conf.all.accept_redirects=0
[root@deep /]# sysctl -w net.ipv4.conf.default.accept_redirects=0
```

Enable bad error message Protection:

This option will alert you about all bad error messages on your network.

Step 1

- Edit the **sysctl.conf** file (vi /etc/sysctl.conf) and add the following line:

```
# Enable/Disable bad error message Protection
net.ipv4.icmp_ignore_bogus_error_responses = 1
```

Step 2

Once configuration has been set, you must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

NOTE: There is another way to update the entry without restarting the network by using the following command into your terminal screen:

```
[root@deep /]# sysctl -w net.ipv4.icmp_ignore_bogus_error_responses=1
```

Enable IP spoofing protection:

The spoofing protection prevents your network from being the source of spoofed (i.e. forged) communications that are often used in DoS Attacks.

Step 1

- Edit the **sysctl.conf** file (vi /etc/sysctl.conf) and add the following lines:

```
# Enable/Disable IP spoofing protection
net.ipv4.conf.all.rp_filter = 2
net.ipv4.conf.default.rp_filter = 2
```

These keys control IP Spoof detection and can have the following values:

- 0=absolutely no checking (the default)
- 1=simple checking (only obvious spoofs)
- 2=strong checking (positive source verification)

The recommended standard is level 2, which can bring "problems in complex (non loop free) networks". In the above parameters, we enable the “strong checking” option.

Step 2
Once the configurations have been made, you must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
```

NOTE: This parameter will prevent spoofing attacks against your internal networks but your external addresses can still be spoofed.

There is another way to update the entry without restarting the network by using the following command into your terminal screen:

```
[root@deep /]# sysctl -w net.ipv4.conf.all.rp_filter=2
[root@deep /]# sysctl -w net.ipv4.conf.default.rp_filter=2
```

Enable Log Spoofed, Source Routed and Redirect Packets:

This change will log all Spoofed Packets, Source Routed Packets, and Redirect Packets to your log files.

Step 1

- Edit the **`sysctl.conf`** file (`vi /etc/sysctl.conf`) and add the following lines:

```
# Enable/Disable Log Spoofed, Source Routed, Redirect Packets
net.ipv4.conf.all.log_martians = 1
net.ipv4.conf.default.log_martians = 1
```

When this key is on (`1`), the kernel will log any "impossible" packets, where the IP source address spoofing is obvious. Example: packet with source address equal to 127.0.0.1 coming from an Ethernet interface. In the above parameter, we enable this option.

Step 2
You must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/init.d/network restart
Setting network parameters    [OK]
Bringing up interface lo      [OK]
Bringing up interface eth0    [OK]
```

NOTE: There is another way to update the entry without restarting the network by using the following command into your terminal screen:

```
[root@deep /]# sysctl -w net.ipv4.conf.all.log_martians=1
[root@deep /]# sysctl -w net.ipv4.conf.default.log_martians=1
```

Other possible optimization of the system

All information described next relates to tuning we can make on the system. Be very careful when attempting this. You can optimize your system, but you can also cause it to crash.

The shared memory limit parameters:

Shared memory is used for inter-process communication, and to store resources that are shared between multiple processes such as cached data and code. If insufficient shared memory is allocated any attempt to access resources that use shared memory such as database connections or executable code will perform poorly.

Under UNIX world, shared memory is referred to as "System V IPC". Almost all modern operating systems provide these features, but not all of them have them turned on or sufficiently sized by default, especially systems with `BSD` heritage. With Linux the default shared memory limit (both `SHMMAX` and `SHMALL`) is 32 MB in 2.4 kernels, but fortunately can be changed into the `/proc` file system.

On system with small `MB` of `RAM` (**>=** 128MB), the default setting of `32MB` for shared memory on the system is **enough** and should not be changed. On system with lot of `RAM`, we can readjust the default setting to better fit our machine and server performance.

Below, I show you an example, to allow `128MB` of shared memory on the system. The new value you enter for the shared memory should be four time less than what your total `MB` of `RAM` is. For example if you have `512MB` of `RAM` installed on your computer, then you can set the default shared memory to `128MB` as we do here. If you have less than what I use in this example, you have to adjust the value to fit your needs.

Step 1
To change the default values of **shared memory**, type the following commands on your terminal:

- Edit the **`sysctl.conf`** file (vi `/etc/sysctl.conf`) and add the following lines:

```
# Improve shared memory size
kernel.shmall = 134217728
kernel.shmmax = 134217728
```

Step 2
You must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

NOTE: There is another way to update the entry without restarting the network by using the following commands into your terminal screen:

```
[root@deep /]# sysctl -w kernel.shmall="134217728"
[root@deep /]# sysctl -w kernel.shmmax="134217728"
```

Tuning the default and maximum window size parameters:

The following hack allow us to raise network limits on Linux by requesting that the kernel provide a larger send buffer for the server's connections to its clients. This is possible by tuning the "`default send window`" and "`maximum send window`" parameters.

Default parameter under Linux is 64kb, this is correct for regular use of the OS but if you run your system as a server, it is recommended to change the default parameter to a sufficiently-high value like 2000kb. 64kb is equal to 65536 (64 * 1024 = 65536), to set the values to 2000kb, we should enter new values of 2048000 (2000 * 1024 = 2048000).

Step 1
To change the default values of **default and maximum window size**, type the following commands on your terminal:

- Edit the **`sysctl.conf`** file (vi `/etc/sysctl.conf)` and add the following lines:

```
# Improve default and maximum window size
net.core.wmem_max = 2048000
net.core.wmem_default = 2048000
```

Step 2
You must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

NOTE: There is another way to update the entry without restarting the network by using the following commands into your terminal screen:

```
[root@deep /]# sysctl -w net.core.wmem_max="2048000"
[root@deep /]# sysctl -w net.core.wmem_default="2048000"
```

The `ulimit` parameter:

The `ulimit` command provides control over the resources available to the shell and to processes started by it, on systems that allow such control. We can use it to change the default resource set by the system to users or super user “root” but actually, we use it for changing resources for the super user “root” only because resources for normal users are controlled and managed through the `/etc/security/limit.conf` file.

It is not all default resources available through the `ulimit` parameter that need to be changed but only those which can improve the performance of the server in a high load environment. Most default values for the super user “root” are acceptable and should be kept unchanged. Linux itself has a “Maximum Processes” per user limit. This feature allows us to control the number of processes an existing user or the “root” super user on the server may be authorized to have. To increase performance on highly loaded servers, we can safely set the limit of processes for the super-user “root” to be unlimited. This is what we will do in the following steps.

One question remains, how can we change the default resources for a specific user on the system? Each new user has a hidden file called "`.bashrc`" available in their home directory. It is into this file that we can change the default value of resources for the specific user. In the example below, we do it for the super user "root" but the procedure is the same for any users on the system. Just edit their corresponding "`.bashrc`" file and make the change.

Step 1

- Edit the **`.bashrc`** file for super user "root" (`vi /root/.bashrc`) and add the line:

```
ulimit -u unlimited
```

NOTE: You must exit and re-login from your terminal for the change to take effect.

Step 2

To verify that you are ready to go, make sure that when you type as root the command **`ulimit -a`** on your terminal, it shows "`unlimited`" next to **`max user processes`**.

```
[root@deep /]# ulimit -a
core file size          (blocks, -c) 0
data seg size           (kbytes, -d) unlimited
file size               (blocks, -f) unlimited
max locked memory       (kbytes, -l) unlimited
max memory size         (kbytes, -m) unlimited
open files                      (-n) 1024
pipe size            (512 bytes, -p) 8
stack size              (kbytes, -s) 8192
cpu time               (seconds, -t) unlimited
max user processes              (-u) unlimited
virtual memory          (kbytes, -v) unlimited
```

NOTE: You may also do `ulimit -u unlimited` at the command prompt instead of adding it to the `/root/.bashrc` file but the value will not survive to a reboot.

The `atime` attribute:

Linux records information about when files were created and last modified as well as when it was last accessed. There is a cost associated with recording the last access time. The `ext2` file system of Linux has an attribute that allows the super-user to mark individual files such that their last access time is not recorded. This may lead to significant performance improvements on often accessed, frequently changing files such as the contents of News Server, Web Server, Proxy Server, Database Server among other directories.

- To set the attribute to a file, use:
 `[root@deep /]# chattr +A filename` ← For a specific file

For a whole directory tree, do something like:

```
[root@deep /root]# chattr -R +A /var/spool          ← For a News and Mail Server directory
[root@deep /root]# chattr -R +A /home/httpd/html    ← For a Web Server directory
[root@deep /root]# chattr -R +A /var/lib/mysql      ← For a SQL Database directory
```

The `noatime` attribute:

Linux has a special mount option for file systems called **`noatime`** that can be added to each line that addresses one file system in the `/etc/fstab` file. If a file system has been mounted with this option, reading accesses to the file system will no longer result in an update to the **`atime`** information associated with the file like we have explained previously. The importance of the **`noatime`** setting is that it eliminates the need by the system to make writes to the file system for files, which are simply being read. Since writes can be somewhat expensive, this can result in measurable performance gains. Note that the write time information to a file will continue to be updated anytime the file is written to. In our example below, we will set the `noatime` option to our `/chroot` file system.

Step 1

- Edit the **`fstab`** file (`vi /etc/fstab`) and add in the line that refers to the `/chroot` file system, the `noatime` option after the `defaults` option as show below:

```
LABEL=/chroot     /chroot     ext3        defaults,noatime    1 2
```

Step 2

Once you have made the necessary adjustments to the `/etc/fstab` file, it is time to inform the system about the modification.

- This can be accomplished with the following commands:

```
[root@deep /]# mount /chroot -oremount
```

Each file system that has been modified must be remounted with the command shown above. In our example we have modified the `/chroot` file system and it is for this reason that we remount this file system with the above command.

Step 3

- You can verify if the modifications have been correctly applied to the Linux system with the following command:

```
[root@deep /]# cat /proc/mounts
/dev/root     /           ext3    rw 0 0
/proc         /proc       proc    rw 0 0
/dev/sda1     /boot       ext3    rw 0 0
/dev/sda10    /cache      ext3    rw,nodev 0 0
/dev/sda9     /chroot     ext3    rw,noatime 0 0
/dev/sda8     /home       ext3    rw,nosuid 0 0
/dev/sda13    /tmp        ext3    rw,noexec,nosuid 0 0
/dev/sda7     /usr        ext3    rw 0 0
/dev/sda11    /var        ext3    rw 0 0
/dev/sda12    /var/lib    ext3    rw 0 0
none          /dev/pts    devpts  rw 0 0
```

This command will show you all file systems on your server with parameters applied to them. If you see something like:

```
/dev/sda9          /chroot    ext3    rw,noatime 0 0
```

Congratulations!



Part III

Network Security

CHAPTER 8

TCP/IP Network Management

IN THIS CHAPTER

1. **`TCP/IP` security problem overview**
2. **Installing more than one Ethernet Card per Machine**
3. **Files-Networking Functionality**
4. **Testing `TCP/IP` Networking**
5. **The last checkup**

Linux TCP/IP

Abstract

This chapter has been inserted here because it is preferable not to be connected to the network if the parts "Installation-Related Reference" and "Security and Optimization-Related Reference" of the book have not been completed. It is not wise to apply new security configurations to your system if you are online. Also, don't forget that the firewall, which represents 50% of networking security, is still not configured on the Linux server. Finally it is very important and I say VERY IMPORTANT that you check all configuration files related to Linux networking to be sure that everything is configured correctly. Please follow all recommendations and steps in this chapter before continuing reading this book. This will allow us to be sure that if something goes wrong in the other chapters, it will be not related to your networking configurations.

- To stop specific network device manually on your system, use the following command:
```
[root@deep /]# ifdown eth0
```

- To start specific network device manually on your system, use the following command:
```
[root@deep /]# ifup eth0
```

- To stop all network devices manually on your system, use the following command:
```
[root@deep /]# /etc/init.d/network stop
Shutting down interface eth0          [OK]
Disabling IPv4 packet forwarding      [OK]
```

- To start all network devices manually on your system, use the following command:
```
[root@deep /]# /etc/init.d/network start
Enabling IPv4 packet forwarding       [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
```

Until now, we have not played with the networking capabilities of Linux. Linux is one of the best operating systems in the world for networking features. Most Internet sites around the world already know this, and have used it for some time. Understanding your network hardware and all the files related to it is very important if you want to have a full control of what happens on your server. Good knowledge of primary networking commands is vital. Network management covers a wide variety of topics. In general, it includes gathering statistical data and monitoring the status of parts of your network, and taking action as necessary to deal with failures and other changes.

The most primitive technique for network monitoring is periodic "pinging" of critical hosts. More sophisticated network monitoring requires the ability to get specific status and statistical information from a range of devices on the network. These should include various sorts of datagram counts, as well as counts of errors of different kinds. For these reasons, in this chapter we will try to answer fundamental questions about networking devices, files related to network functionality, and essential networking commands.

TCP/IP security problem overview

It is assumed that the reader is familiar with the basic operation of the TCP/IP protocol suite, which includes IP and TCP header field functions and initial connection negotiation. For the uninitiated, a brief description of TCP/IP connection negotiation is given below. The user is strongly encouraged however to research other published literature on the subject.

The IP Packets:

The term packet refers to an **I**nternet **P**rotocol (IP) network message. It's the name given to a single, discrete message or piece of information that is sent across an Ethernet network. Structurally, a packet contains an information header and a message body containing the data being transferred. The body of the IP packet- it's data- is all or a piece (a fragment) of a higher-level protocol message.

The IP mechanism:

Linux supports three IP message types: ICMP, UDP, and TCP. An ICMP (**I**nternet **C**ontrol **M**essage **P**rotocol) packet is a network-level, IP control and status message.

ICMP messages contains information about the communication between the two end-point computers.

A UDP (**U**ser **D**atagram **P**rotocol) IP packet carries data between two network-based programs, without any guarantees regarding successful delivery or packet delivery ordering. Sending a UDP packet is akin to sending a postcard to another program.

A TCP (**T**ransmission **C**ontrol **P**rotocol) IP packet carries data between two network-based programs, as well, but the packet header contains additional state information for maintaining an ongoing, reliable connection. Sending a TCP packet is akin to carrying on a phone conversation with another process. Most Internet network services use the TCP communication protocol rather than the UDP communication protocol. In other words, most Internet services are based on the idea of an ongoing connection with two-way communication between a client program and a server program.

The IP packet headers:

All IP packet headers contain the source and destination IP addresses and the type of IP protocol message (ICMP, UDP or TCP) this packet contains. Beyond this, a packet header contains slightly different fields depending on the protocol type. ICMP packets contain a type field identifying the control or status message, along with a second code field for defining the message more specifically. UDP and TCP packets contain source and destination service port numbers. TCP packets contain additional information about the state of the connection and unique identifiers for each packet.

The TCP/IP Security Problem:

The TCP/IP protocol suite has a number of weaknesses that allows an attacker to leverage techniques in the form of covert channels to surreptitiously pass data in otherwise benign packets. This section attempts to illustrate these weaknesses in theoretical examples.

Application:

A covert channel is described as: "any communication channel that can be exploited by a process to transfer information in a manner that violates the systems security policy. Essentially, it is a method of communication that is not part of an actual computer system design, but can be used to transfer information to users or system processes that normally would not be allowed access to the information.

In the case of TCP/IP, there are a number of methods available whereby covert channels can be established and data can be surreptitiously passed between hosts. These methods can be used in a variety of areas such as the following:

- ✓ Bypassing packet filters, network sniffers, and "dirty word" search engines.
- ✓ Encapsulating encrypted or non-encrypted information within otherwise normal packets of information for secret transmission through networks that prohibit such activity "TCP/IP Steganography".
- ✓ Concealing locations of transmitted data by "bouncing" forged packets with encapsulated information off innocuous Internet sites.

It is important to realize that TCP is a "connection oriented" or "reliable" protocol. Simply put, TCP has certain features that ensure data arrives at the remote host in a usually intact manner. The basic operation of this relies in the initial TCP "three way handshake" which is described in the three steps below.

Step 1
Send a synchronize (**SYN**) packet and Initial Sequence Number (**ISN**)

Host A wishes to establish a connection to Host B. Host A sends a solitary packet to Host B with the synchronize bit (**SYN**) set announcing the new connection and an Initial Sequence Number (**ISN**) which will allow tracking of packets sent between hosts:

```
Host A  ------  SYN(ISN) ------>     Host B
```

Step 2
Allow remote host to respond with an acknowledgment (**ACK**)

Host B responds to the request by sending a packet with the synchronize bit set (**SYN**) and **ACK** (acknowledgment) bit set in the packet back to the calling host. This packet contains not only the responding clients' own sequence number, but the Initial Sequence Number plus one (**ISN+1**) to indicate the remote packet was correctly received as part of the acknowledgment and is awaiting the next transmission:

```
Host A  <------  SYN(ISN+1)/ACK ------     Host B
```

Step 3
Complete the negotiation by sending a final acknowledgment to the remote host.

At this point Host A sends back a final **ACK** packet and sequence number to indicate successful reception and the connection is complete and data can now flow:

```
Host A  ------ ACK ------>     Host B
```

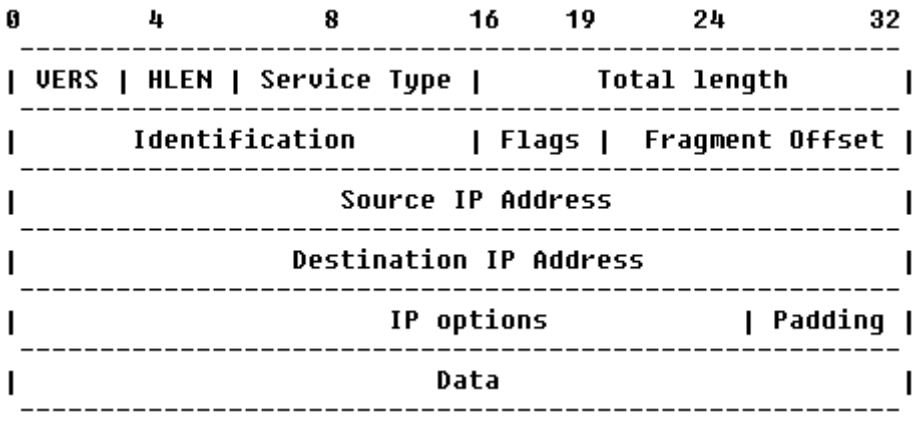
The entire connection process happens in a matter of milliseconds and both sides independently acknowledge each packet from this point. This handshake method ensures a "reliable" connection between hosts and is why TCP is considered a "connection oriented" protocol.

It should be noted that only TCP packets exhibit this negotiation process. This is not so with UDP packets which are considered "unreliable" and do not attempt to correct errors nor negotiate a connection before sending to a remote host.

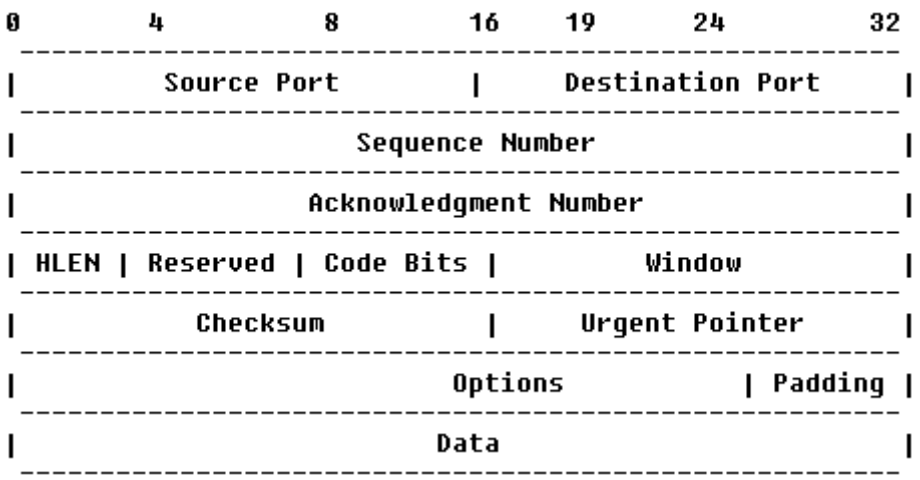
Encoding Information in a TCP/IP Header:

The TCP/IP header contains a number of areas where information can be stored and sent to a remote host in a covert manner. Take the following diagrams, which are textual representations of the IP and TCP headers respectively:

IP Header (Numbers represent bits of data from 0 to 32 and the relative position of the fields in the datagram)



TCP Header (Numbers represent bits of data from 0 to 32 and the relative position of the fields in the datagram)



Within each header there are multitudes of areas that are not used for normal transmission or are "optional" fields to be set as needed by the sender of the datagrams. An analysis of the areas of a typical IP header that are either unused or optional reveals many possibilities where data can be stored and transmitted.

The basis of the exploitation relies in encoding ASCII values of the range 0-255. Using this method it is possible to pass data between hosts in packets that appear to be initial connection requests, established data streams, or other intermediate steps. These packets can contain no actual data, or can contain data designed to look innocent. These packets can also contain forged source and destination IP addresses as well as forged source and destination ports.

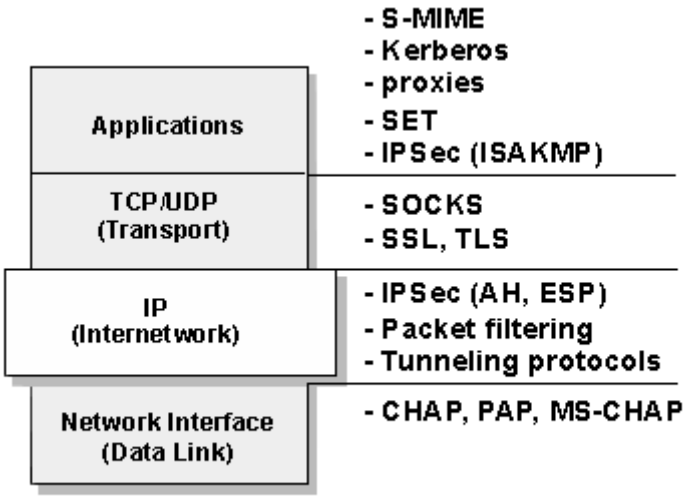
This can be useful for tunneling information past some types of packet filters. Additionally, forged packets can be used to initiate an anonymous TCP/IP "bounced packet network" whereby packets between systems can be relayed off legitimate sites to thwart tracking by sniffers and other network monitoring devices.

Implementations of Security Solutions:

The following protocols and systems are commonly used to solve and provide various degrees of security services in a computer network.

- IP filtering
- Network Address Translation (NAT)
- IP Security Architecture (IPSec)
- SOCKS
- Secure Sockets Layer (SSL)
- Application proxies
- Firewalls
- Kerberos and other authentication systems (AAA servers)
- Secure Electronic Transactions (SET)

This graph illustrates where those security solutions fit within the TCP/IP layers:



Security Solutions in the TCP/IP Layers

Installing more than one Ethernet Card per Machine

You might use Linux as a gateway between two Ethernet networks. In that case, you might have two Ethernet cards on your server. To eliminate problems at boot time, the Linux kernel doesn't detect multiple cards automatically. If you happen to have two or more cards, you should specify the parameters of the cards in the `lilo.conf` file for a Monolithic kernel or in the `modules.conf` file for a Modularized kernel. The following are problems you may encounter with your network cards.

Problem 1
If the driver(s) of the card(s) is/are being used as a loadable module (Modularized kernel), in the case of PCI drivers, the module will typically detect all of the installed cards automatically. For ISA cards, you need to supply the I/O base address of the card so the module knows where to look. This information is stored in the file `/etc/modules.conf`.

As an example, consider we have two ISA 3c509 cards, one at I/O 0x300 and one at I/O 0x320.

- For ISA cards, edit the **`modules.conf`** file (`vi /etc/modules.conf`) and add:

```
alias eth0 3c509
alias eth1 3c509
options 3c509 io=0x300,0x320
```

This says that the 3c509 driver should be loaded for either eth0 or eth1 (alias eth0, eth1) and it should be loaded with the options io=0x300,0x320 so that the drivers knows where to look for the cards. Note that 0x is important – things like 300h as commonly used in the DOS world won't work.

For PCI cards, you typically only need the alias lines to correlate the ethN interfaces with the appropriate driver name, since the I/O base of a PCI card can be safely detected.

- For PCI cards, edit the **`modules.conf`** file (`vi /etc/modules.conf`) and add:

```
alias eth0 3c509
alias eth1 3c509
```

Problem 2
If the drivers(s) of the card(s) is/are compiled into the kernel (Monolithic kernel), the PCI probes will find all related cards automatically. ISA cards will also find all related cards automatically, but in some circumstance ISA cards still need to do the following. This information is stored in the file `/etc/lilo.conf`. The method is to pass boot-time arguments to the kernel, which is usually done by LILO.

- For ISA cards, edit the **`lilo.conf`** file (`vi /etc/lilo.conf`) and add:

```
append="ether=0,0,eth1"
```

In this case eth0 and eth1 will be assigned in the order that the cards are found at boot. Remember that this is required only in some circumstance for ISA cards, PCI cards will be found automatically.

NOTE: First test your ISA cards without the boot-time arguments in the `lilo.conf` file, and if this fails, use the boot-time arguments.

Files-Networking Functionality

In Linux, the TCP/IP network is configured through several text files. You may have to edit them to make the network work. It's very important to know the configuration files related to TCP/IP networking, so that you can edit and configure the files if necessary. Remember that our server doesn't have an Xwindow interface (GUI) to configure files via a graphical interface. Even if you use a GUI in your daily activities it is important to know how to configure the network configuration files in text mode. The following sections describe all the basic TCP/IP configuration files under Linux.

The /etc/sysconfig/network-scripts/ifcfg-ethN files:

The configuration files for each network device you may have or want to add on your system are located in the /etc/sysconfig/network-scripts directory with Red Hat Linux, and are named **ifcfg-eth0** for the first interface and **ifcfg-eth1** for the second, etc. It is recommended to verify if all the parameters in this file are correct.

Following is a sample /etc/sysconfig/network-scripts/ifcfg-eth0 file:

```
DEVICE=eth0
BOOTPROTO=static
BROADCAST=208.164.186.255
IPADDR=208.164.186.1
NETMASK=255.255.255.0
NETWORK=208.164.186.0
ONBOOT=yes
USERCTL=no
```

If you want to modify your network address manually, or add a new one on a new interface, edit this file (ifcfg-ethN), or create a new one and make the appropriate changes.

DEVICE=***devicename***, where ***devicename*** is the name of the physical network device.

BOOTPROTO=***proto***, where ***proto*** is one of the following:

- static - The default option of Linux (static IP address) sould be used.
- none - No boot-time protocol should be used.
- bootp - The bootp (now pump) protocol should be used.
- dhcp - The dhcp protocol should be used.

BROADCAST=***broadcast***, where ***broadcast*** is the broadcast IP address.

IPADDR=***ipaddr***, where ***ipaddr*** is the IP address.

NETMASK=***netmask***, where ***netmask*** is the netmask IP value.

NETWORK=***network***, where ***network*** is the network IP address.

ONBOOT=***answer***, where ***answer*** is yes or no (Does the interface will be active or inactive at boot time).

USERCTL=***answer***, where ***answer*** is one of the following:

- yes (Non-root users are allowed to control this device).
- no (Only the super-user root is allowed to control this device).

The /etc/resolv.conf file:

This file /etc/resolv.conf is another text file, used by the resolver—a library that determines the IP address for a host name. It is recommended to verify if all parameters included in this file are correct.

Following is a sample /etc/resolv.conf file:

```
domain openna.com
search ns1.openna.com ns2.openna.com openna.com
nameserver 208.164.186.1
nameserver 208.164.186.2
nameserver 127.0.0.1
```

NOTE: Name servers are queried in the order they appear in the file (primary, secondary).

The /etc/host.conf file:

This file /etc/host.conf specifies how names are resolved. Linux uses a resolver library to obtain the IP address corresponding to a host name. It is recommended to verify that all parameters included in this file are correct.

Following is a sample /etc/host.conf file:

```
# Lookup names via /etc/hosts first then fall back to DNS resolver.
order hosts,bind
# We have machines with multiple addresses.
multi on
```

The **order** option indicates the order of services. The sample entry specifies that the resolver library should first consult the /etc/hosts file of Linux to resolve a name and then check the name server (DNS).

The **multi** option determines whether a host in the /etc/hosts file can have multiple IP addresses (multiple interface ethN). Hosts that have more than one IP address are said to be *multihomed*, because the presence of multiple IP addresses implies that the host has several network interfaces.

The /etc/sysconfig/network file:

The /etc/sysconfig/network file is used to specify information about the desired network configuration on your server. It is recommended that you verify all the parameters included in this file are correct.

Following is a sample /etc/sysconfig/network file:

```
NETWORKING=yes
HOSTNAME=deep
GATEWAY=207.35.78.1
GATEWAYDEV=eth0
```

The following values may be used:

NETWORKING=***answer***, where ***answer*** is yes or no (Configure networking or not configure networking).

HOSTNAME=***hostname***, where ***hostname*** is the hostname of your server.

GATEWAY=***gwip***, where ***gwip*** is the IP address of the remote network gateway (if available).

GATEWAYDEV=***gwdev***, where ***gwdev*** is the device name (eth#) you use to access the remote gateway.

The `/etc/sysctl.conf` file:

With the new version of Red Hat Linux, all kernel parameters available under the `/proc/sys/` subdirectory of Linux can be configured at runtime. You can use the new `/etc/sysctl.conf` file to modify and set kernel parameters at runtime. The `sysctl.conf` file is read and loaded each time the system reboots or when you restart your network. All settings are now stored in the `/etc/sysctl.conf` file. All modifications to `/proc/sys` should be made through `/etc/sysctl.conf`, because they are better for control, and are executed before `rc.local` or any other "users" scripts.

Below, we'll focus only on the kernel option for `IPv4` forwarding support. See later in this chapter the `TCP/IP` security parameters related to the `sysctl.conf` file.

To enable `IPv4` forwarding on your Linux system, use the following command:

Step 1

- Edit the `sysctl.conf` file (`vi /etc/sysctl.conf`) and add the following line:

```
# Enable packet forwarding (required only for Gateway, VPN, Proxy, PPP)
net.ipv4.ip_forward = 1
```

Step 2

You must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

WARNING: You must enable packet forwarding only on a machine that serves as a Gateway Server, `VPN` Server, Proxy Server or with `PPP` connection. Forwarding allows packets that are destined for another network interface (if you have another one) to pass through the network.

There is another way to update the entry without restarting the network by using the following command into your terminal screen:

```
[root@deep /]# sysctl -w net.ipv4.ip_forward=1
```

The /etc/hosts file:

As your machine gets started, it will need to know the mapping of some hostnames to IP addresses before DNS can be referenced. This mapping is kept in the /etc/hosts file. In the absence of a name server, any network program on your system consults this file to determine the IP address that corresponds to a host name.

Following is a sample /etc/hosts file:

IP Address	Hostname	Alias
127.0.0.1	localhost.localdomain	localhost
208.164.186.1	deep.openna.com	deep
208.164.186.2	mail.openna.com	mail
208.164.186.3	web.openna.com	web

The leftmost column is the IP address to be resolved. The next column is that host's name. Any subsequent columns are the aliases for that host. In the second line, for example, the IP address 208.164.186.1 if for the host deep.openna.com. Another name for deep.openna.com is deep.

WARNING: Some people have reporting that a badly formed line in the /etc/hosts file may result to a "Segmentation fault (core dumped)" with the syslogd daemon, therefore I recommend you to double check your entry under this file and be sure that its respond to the example as shown above. The "Alias" part of the line is important if you want to be able to use the FQDN (**F**ully **Q**ualified **D**omain **N**ame) of the system reported by the hostname -f command.

After you are finished adding and configuring your networking files, don't forget to restart your network for the changes to take effect.

- To restart your network, use the following command:

```
[root@deep /]# /etc/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

WARNING: Time out problems for telnet or ftp connection are often caused by the server trying to resolve the client IP address to a DNS name. Either DNS isn't configured properly on your server or the client machines aren't known to the DNS server. If you intend to run telnet or ftp services on your server, and aren't using DNS, don't forget to add the client machine name and IP in your /etc/hosts file on the server or you can expect to wait several minutes for the DNS lookup to time out, before you get a login prompt.

Testing TCP/IP Networking

Once we have applied TCP/IP security and optimization parameters to our server and checked or configured all files related to network functionality, we can run some tests to verify that everything works as expected. It is important to note that at this stage every test must be successful and not have any errors. It is to your responsibility to know and understand networking architecture and basic TCP/IP protocols before testing any parts of your networking configuration and topology.

Step 1
To begin, we can use the **ifconfig** utility to display all the network interfaces on the server.

- To display all the interfaces you have on your server, use the command:
 `[root@deep /]# ifconfig`

 The output should look something like this:

```
eth0  Link encap:Ethernet  HWaddr 00:E0:18:90:1B:56
      inet addr:208.164.186.2  Bcast:208.164.186.255  Mask:255.255.255.0
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:1295 errors:0 dropped:0 overruns:0 frame:0
      TX packets:1163 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:100
      Interrupt:11 Base address:0xa800

 lo   Link encap:Local Loopback
      inet addr:127.0.0.1  Mask:255.0.0.0
      UP LOOPBACK RUNNING  MTU:3924  Metric:1
      RX packets:139 errors:0 dropped:0 overruns:0 frame:0
      TX packets:139 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
```

NOTE: If the **ifconfig** tool is invoked without any parameters, it displays all interfaces you configured. An option of "**-a**" shows the inactive one as well.

Step 2
If all network interfaces on the server look as you expect, then it is time to verify that you can reach your hosts. Choose a host from your internal network, for instance 192.168.1.1

- To verify that you can reach your internal hosts, use the command:
 `[root@deep /]# ping 192.168.1.1`

 The output should look something like this:

```
PING 192.168.1.1 (192.168.1.1) from 192.168.1.1 : 56 data bytes
64 bytes from 192.168.1.1: icmp_seq=0 ttl=128 time=1.0 ms
64 bytes from 192.168.1.1: icmp_seq=1 ttl=128 time=1.0 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=128 time=1.0 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=128 time=1.0 ms

--- 192.168.1.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 1.0/1.0/1.0 ms
```

WARNING: Do not try to `ping` a host in which you have applied the previous `TCP/IP` security settings to prevent your system to respond to `ping` request. Instead try to `ping` another host without this feature enable. Also if you don't receive an answer from the internal host you try to `ping`, verify if your hubs, routers, network cards, and network topology are correct.

If you are able to **`ping`** your internal host, congratulations! Now we must **`ping`** an external network, for instance `216.148.218.195`

- To verify that you can reach the external network, use the command:
 `[root@deep /]#` **`ping 216.148.218.195`**

 The output should look something like this:

```
PING 216.148.218.195 (216.148.218.195) from 216.148.218.195 :56 data byte
64 bytes from 216.148.218.195: icmp_seq=0 ttl=128 time=1.0 ms
64 bytes from 216.148.218.195: icmp_seq=1 ttl=128 time=1.0 ms
64 bytes from 216.148.218.195: icmp_seq=2 ttl=128 time=1.0 ms
64 bytes from 216.148.218.195: icmp_seq=3 ttl=128 time=1.0 ms

--- 216.148.218.195 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 1.0/1.0/1.0 ms
```

Step 3
You should now display the routing information with the command **`route`** to see if the hosts have the correct routing entries.

- To display the routing information, use the command:
 `[root@deep /]#` **`route -n`**

The output should look something like this:

```
Kernel IP routing table
Destination   Gateway      Genmask        Flags  Metric  Ref    Use   Iface
208.164.186.2 0.0.0.0      255.255.255.255UH     0       0      0     eth0
208.164.186.0 208.164.186.2 255.255.255.0  UG     0       0      0     eth0
208.164.186.0 0.0.0.0      255.255.255.0  U      0       0      0     eth0
127.0.0.0     0.0.0.0      255.0.0.0      U      0       0      0     lo
```

Step 4
Another useful option is "**`netstat -vat`**", which shows all active and listen `TCP` connections.

- To shows all active and listen `TCP` connections, use the command:
 `[root@deep /]#` **`netstat -vat`**

The output may look something similar to this example depending if the related services are running. Be aware that your results will almost certainly vary from the ones shown below:

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address     State
tcp        0      0 deep.openna.co:domain   *:*                 LISTEN
tcp        0      0 localhost:domain        *:*                 LISTEN
tcp        0      0 deep.openna.com:ssh     gate.openna.com:1682ESTABLISHED
tcp        0      0 *:webcache              *:*                 LISTEN
tcp        0      0 deep.openar:netbios-ssn *:*                 LISTEN
tcp        0      0 localhost:netbios-ssn   *:*                 LISTEN
tcp        0      0 localhost:1032          localhost:1033      ESTABLISHED
tcp        0      0 localhost:1033          localhost:1032      ESTABLISHED
tcp        0      0 localhost:1030          localhost:1031      ESTABLISHED
tcp        0      0 localhost:1031          localhost:1030      ESTABLISHED
tcp        0      0 localhost:1028          localhost:1029      ESTABLISHED
tcp        0      0 localhost:1029          localhost:1028      ESTABLISHED
tcp        0      0 localhost:1026          localhost:1027      ESTABLISHED
tcp        0      0 localhost:1027          localhost:1026      ESTABLISHED
tcp        0      0 localhost:1024          localhost:1025      ESTABLISHED
tcp        0      0 localhost:1025          localhost:1024      ESTABLISHED
tcp        0      0 deep.openna.com:www     *:*                 LISTEN
tcp        0      0 deep.openna.com:https   *:*                 LISTEN
tcp        0      0 *:389                   *:*                 LISTEN
tcp        0      0 *:ssh                   *:*                 LISTEN
```

Step 5
Sometimes machines on your network will discard your `IP` packets and finding the offending Gateway responsilbe can be difficult. Fortunately the **`tracepath`** utility attempts to trace the route an `IP` packet would follow to some Internet host. Choose an Internet host, for instance `64.81.28.146`

- To print the route packets take to network host, use the command:
 `[root@deep /]#` **`tracepath 64.81.28.146`**

The output should look something like this:

```
1?: [LOCALHOST]      pmtu 1500
 1?: 207.35.78.1
 2?: 10.70.1.1
 3?: 206.47.228.178
 4?: 206.108.97.149
 5?: 206.108.103.214
 6?: 206.108.103.228
 7?: 208.51.134.9
 8?: 208.48.234.189
 9?: 206.132.41.78    asymm 10
10?: 204.246.213.226  asymm 13
11?: 206.253.192.217  asymm 13
12?: 206.253.195.218  asymm 14
13:  64.81.28.146     asymm 15 139ms reached
     Resume: pmtu 1500 hops 13 back 15
```

Step 6
Finally, we will use the `hostname` command of Linux to show if our systems host name is correct.

- To display and print the current host name of your server, use the command:
```
[root@deep /]# hostname
deep
```

The `hostname` command without any options will print the current host name of our system, in this example "`deep`".

Now, it's important to verify if the **F**ully **Q**ualified **D**omain **N**ame (`FQDN`) of our server is reported correctly.

- To display and print the `FQDN` of your server, use the command:
```
[root@deep /]# hostname -f
deep.openna.com
```

The last checkup

If you can answer, "Yes" to each of the questions below, then your network is working and you can continue .

- ✓ Parameters inside `ifcfg-ethN` files are corrects
- ✓ The `/etc/resolv.conf` file contain your primary and secondary Domain Name Server
- ✓ All parameters included in the `/etc/host.conf` file are corrects
- ✓ All parameters included in the `/etc/sysconfig/network` file are corrects
- ✓ The `/etc/hosts` file contain the mapping of your hostnames to IP addresses
- ✓ All network interfaces on the server have the right parameter
- ✓ You can reach the internal and external hosts
- ✓ Your hosts have the correct routing entry
- ✓ The status of the interfaces has been checked and looks fine
- ✓ You are able to print the route packets take to network host

CHAPTER 9

Firewall Basic Concept

IN THIS CHAPTER

1. What is the IANA?
2. The ports numbers
3. What is a Firewall?
4. Packet Filter vs. Application Gateway
5. What is a Network Firewall Security Policy?
6. The Demilitarized Zone
7. Linux `IPTables` Firewall Packet Filter
8. The Netfilter Architecture

Linux Firewall

Abstract
Before going into the installation, configuration and use of firewall software with Linux, we have to explain a little bit about what a firewall is, how it works and how this effects into your network and servers. A firewall is the first line of defense for your system, it is the first place where network connections and contacts will appear on the server before any server services or programs are started for them.

What is the IANA?
The **I**nternet **A**ssigned **N**umbers **A**uthority (IANA) is the central coordinator for the assignment of unique parameter values for Internet protocols. The IANA is chartered by the **I**nternet **Soc**iety (ISOC), and the **F**ederal **N**etwork **C**ouncil (FNC), to act as the clearinghouse to assigning and coordinating the use of the numerous Internet protocol parameters.

The Internet protocol suite, as defined by the **I**nternet **E**ngineering **T**ask **F**orce (IETF) and its steering group (the IESG), contains numerous parameters, such as internet addresses, domain names, autonomous system numbers (used in some routing protocols), protocol numbers, port numbers, management information base object identifiers, including private enterprise numbers, and many others.

The common use of Internet protocols by the Internet community requires that the particular values used in these parameter fields be assigned UNIQUELY. It is the task of the IANA to make these unique assignments, as requested, and to maintain a registry of the currently assigned values.

As an example, imagine that you have developed a new networking program that runs as a daemon on the server and it requires a port number. It is up to the IANA to register, manage and maintain a unique port number dedicated for and associated with your program. This way, anyone that wants to use your program, will know which unique port number is associated with it.

The ports numbers
In order for a computer to connect to multiple Internet services at the same time, the concept of a 'port' was introduced. Each computer has 65535 ports available. If your web browser initiates a connection to www.openna.com (port 80 by default) for example, it will pick the first available port (lets say 10232) and use it to send the connection request to www.openna.com. Openna.com's web server will reply to port 10232 on your PC. This way, your PC knows that this reply is in response to the request sent to www.openna.com earlier. All open ports should have a service or daemon running on them. A service or daemon is simply the software running on these ports, which provides a service to the users who connect to it. If no service or daemon is running on the port, then there is no reason to have the port open on the server and you should close it.

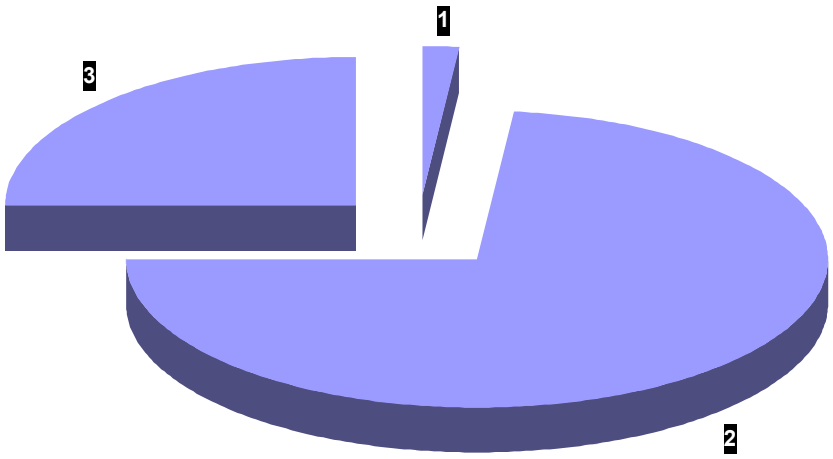
The port numbers are divided into three ranges: the Well Known Ports, the Registered Ports, and the Dynamic and/or Private Ports. There are two types of ports, using two different protocols: TCP and UDP. Although they are different protocols, they can have the same port number. The Well Known Ports are those from 0 through 1023, the Registered Ports are those from 1024 through 49151 and the Dynamic and/or Private Ports are those from 49152 through 65535.

The Well Known Ports are assigned by the IANA and on most systems can only be used by system (or root) processes or by programs executed by privileged users (our daemons running in the background). Ports are used by the TCP protocol [RFC793] to name the ends of logical connections, which carry long-term conversations. For the purpose of providing services to unknown callers, a service contact port is defined. The contact port is sometimes called the "well-known port". Wherever possible, the same port assignments are also used by UDP protocol [RFC768]. For many years Well Known Ports were in the range 0-255. Recently, the range for the Well Known Ports has been expanded from 0 to 1023 to respond to the exponential growth of the Internet.

The Registered Ports are also listed by the IANA and, on most systems, can be used by ordinary user processes or programs executed by ordinary users. The IANA registers the use of these ports as a convenience to the community. Again, wherever possible, these same port assignments are used with UDP [RFC768]. The Registered Ports are in the range 1024-49151.

Finally, the Dynamic and/or Private Ports are those ranging from 49152 through to 65535.

Ports Numbers Graphical Representation



[1] The Well Known Ports represent 2% of all available ports [0-1023].
[2] The Registered Ports represent 73% of all available ports [1024-49151].
[3] The Dynamic and/or Private Ports represent 25% of all available ports [49152-65535].

What is a Firewall?

As we said before, if a service or daemon program is not running on its assigned port, then there is no reason to have the related port open on the server.

There are some simple reasons why this is bad:

1. We can run a Trojan program on the open port.
2. We can use the open port to access another server.

A firewall (software or hardware) will take care of this. It will close all ports that we don't use on the server. Firewalls can control, manage and supervise all legitimate open ports where services or daemons are running. To recap, an Internet firewall is a software program or hardware device used to protect a server or private network from the ever-raging fire on the Internet.

The best practice is to run a firewall on each server, even if you have a router or a big firewall in front of your other servers on the network. This allows us to close any open ports that we don't use, and to better control what goes in and out of our servers and add another level of security to our network.

Packet Filter vs. Application Gateway

During the past ten-years, different firewall technologies have been developed to respond to different server's requirements on the Internet. From this research two distinct categories of firewall software have emerged. The first category of firewall is known as **Packet Filtering** and the second as an **Application Gateway**. It is important to make the distinction between the two categories before continuing. This will allow us to understand the technology used in each one and to get a better idea about where we need to use them on our servers and network. Each one has its advantages and this is one of the reasons why both categories still exist. It is up to us to use the right firewall software depending of the kind of services that we want to offer in order to protect our Linux servers and network.

Packet Filtering

Packet Filtering is the type of firewall that's built into the Linux kernel (as a kernel module, or compiled in). A filtering firewall works at the network level. Data is only allowed to leave the system if the firewall rules allow it. As packets arrive they are filtered by their type, source address, destination address, and port information contained in each packet header.

Most of the time, packet filtering is accomplished by using a router that can forward packets according to filtering rules. When a packet arrives at the packet-filtering router, the router extracts certain information from the packet header and makes decisions according to the filter rules as to whether the packet will be allowed to pass through or be discarded.

The following information can be extracted from the packet header:

- ✓ Source IP address
- ✓ Destination IP address
- ✓ TCP/UDP source port
- ✓ TCP/UDP destination port
- ✓ ICMP message type
- ✓ Encapsulated protocol information (TCP, UDP, ICMP or IP tunnel)

Because very little data is analyzed and logged, filtering firewalls take less CPU power and create less latency in your network. Two generations of Packet Filtering Firewall software have been made available to the public.

The first generation was called "static", because the method of connecting between the internal and external networks must be left open at all times. Static Packet filtering Firewall (first generation) is well known under Linux as the `IPCHAINS` firewall software used in the Linux Kernel version 2.2.x. The main disadvantage of this type of firewall is the fact that ports must be left open at all times to allow desired traffic, another important disadvantage is that it allows a direct connection to internal hosts by external clients, and finally it offers no user authentication.

To address some of the problems of the first generation of Packet filtering Firewalls, a second generation of Packet Filtering software was developed. The second generation is known as Dynamic Packet Filters or Stateful Packet Filtering also known under Linux as `IPTables` firewall software, used in Linux Kernel version 2.4.x. The stateful packet filter keeps track of the state and context information of a session. Once a series of packets has passed through the "door" to it's destination, the firewall closes the door. This solves the problem of having ports open at all times. Another improvement compared to the first generation is the limitation of spoofing attacks. Dynamic Packet Filters is not perfect and external systems are still able to make an IP connection with an internal host and user authentication still not supported.

Application Gateways
An Application Gateway, also known as proxy software and well known under Linux as "`Squid`" software, is a firewall system in which processes that provide services maintain complete TCP connection states and sequencing. At this time two generations of Application Gateway Firewall software have been made available to the public. The first generation was simply called an "Application Gateway". With this type of firewall software, all connections to the internal network go through the firewall for verification and approval, based on the set-up policies that you have entered in the configuration file of the Application Gateway Firewall. Contrary to a Packet Filtering Firewall, an Application Gateway Firewall looks in detail at the communication stream before allowing the traffic to pass into the network to its final destination by analyzing application commands inside the payload portion of data packets. Whereas stateful packet filters systems do not. Another important advantage of an Application Gateway is the fact that it does not allow any direct connections between internal and external hosts and it also supports user-level authentication, two points where packet filter lose again.

But, Application Gateway Firewall software is not perfect and has some bad points too. The first is that it is slower than packet filtering, it requires that the internal client (i.e. the workstation) to knows about them and it also does not support every type of connection.

To address some of the problems encountered in the first generation of this type of firewall software, a second generation has been developed. It's called the Transparent Application Gateway and one of its main advantage compared to its predecessor is that client workstations do not either have to be aware of the firewall nor run special software to communicate with the external network. This fixes the problem of having the internal client (i.e. the workstation) know about them. Even with all these improvement, some disadvantages still exist. Transparent Application Gateways are slower than packet filters, they consume more system resources and do not support every type of connection.

From the above analysis (Packet Filter vs. Application Gateway), we can summarize the main advantages and disadvantages of each firewall category as follows:

Packet Filter advantages are:

- ✓ Good for traffic management.
- ✓ Low Overhead / High Throughput.
- ✓ Supports almost any service.

Application Gateway advantages are:

- ✓ Do not allow any direct connections between internal and external hosts.
- ✓ Support user-level authentication.

Packet Filter disadvantages are:

- ✓ Offers no user authentication.
- ✓ Allows direct IP connections to internal hosts by external clients.

Application Gateway disadvantages are:

- ✓ Slower than packet filters.
- ✓ Do not support every possible type of connection.

Therefore we can, with confidence, recommend Packet Filter Firewall software for all servers in the DMZ zone (The Demilitarized Zone) under a Unix environment. For Windows systems, the approach is not recommended, implementations and strategies are different due to the insecure nature of the operating system and it's programs. Unix systems and their programs have many features to compensate some of the disadvantages of Packet Filter Firewalls and this is the reason why this type of firewall does not pose any problems for Unix systems located in the DMZ like web, mail, ftp, lists, virtual, dns, database, and backup servers.

An Application Gateway Firewall is recommended only for a Gateway Server (a machine that makes a bridge between your private internal network and the Internet). Also a Packet Filter Firewall is recommended for Gateway servers and this means that you have to install an Application Gateway Firewall and a Packet Filter Firewall on a Gateway Server. Yes, both are recommended for a secure communication between your private internal hosts and the Internet. Using just one type of firewall on a Gateway Server is not enough.

Finally, I will say that installing an Application Gateway Firewall on web, mail, ftp, lists, virtual, dns, database, and backup servers is a waste of time. You only need this kind of firewall software on a Gateway Server.

What is a Network Firewall Security Policy?

A network firewall security policy defines those services that will be explicitly allowed or denied, how these services will be used and the exceptions to these rules. An organization's overall security policy must be determined according to a security and business-needs analysis. Since a firewall relates to network security alone, a firewall has little value unless the overall security policy is properly defined. Every rule in the firewall security policy should be implemented on a firewall. Generally, a firewall uses one of the following methods.

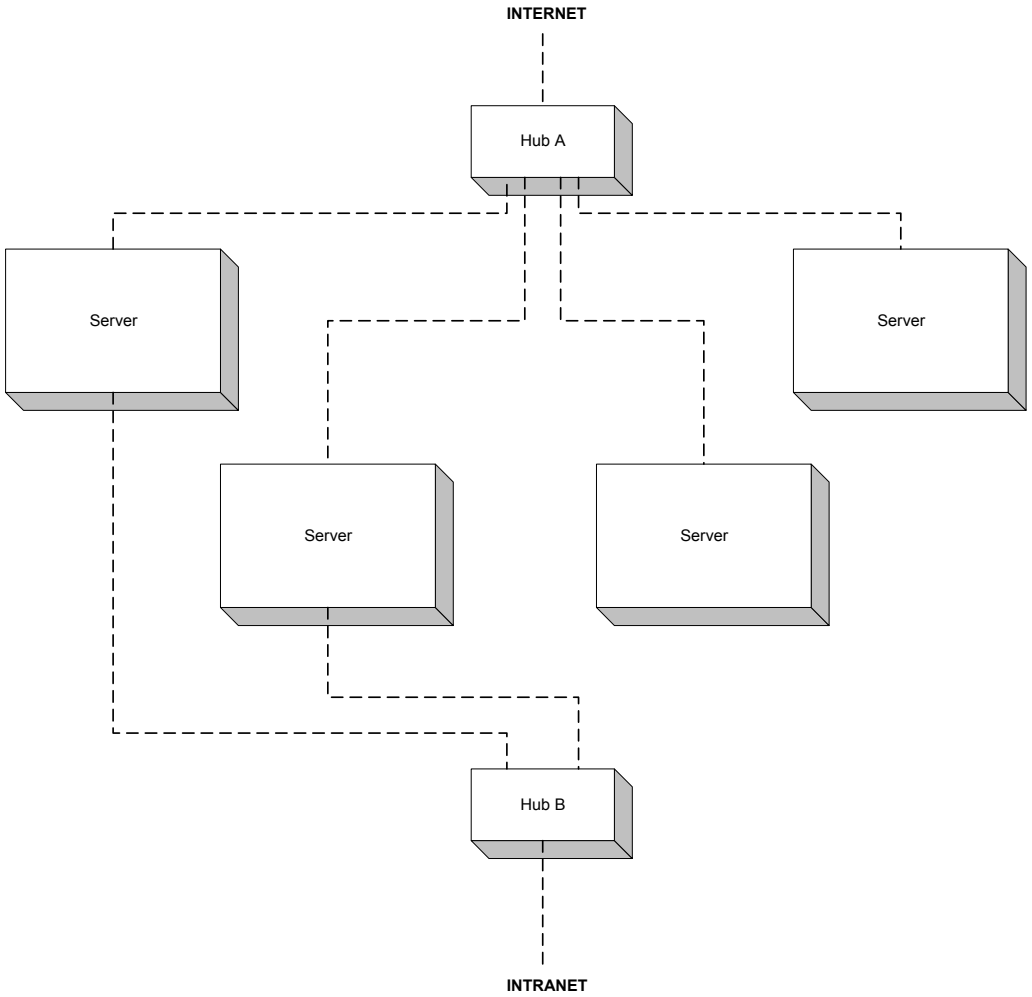
Everything not specifically permitted is denied

This approach blocks all traffic between two networks except for those services and applications that are permitted. Therefore, each required service or application should be implemented on an individual basis. No service or application that could possibly be a potential hole on the firewall should be permitted. This is the most secure method of firewalling, denying services and applications unless explicitly allowed by the administrator. On the other hand, from the users point of view, it might be more restrictive and less convenient. This is the method we will use in our Firewall configuration files in this book.

Everything not specifically denied is permitted
This approach allows all traffic between two networks except for those services and applications that are denied. Therefore, each untrusted or potentially harmful service or application should be denied individually. Although this is flexible and convenient for the users, it could potentially cause some serious security problems. This method is really not recommended.

The Demilitarized Zone

A demilitarized zone (DMZ) refers to a part of the network that is neither part of the internal network nor directly part of the Internet. Typically, this is the area between your Internet access router and your bastion host (internal network), though it can be between any two policy-enforcing components of your architecture. A DMZ minimizes the exposure of hosts on your external LAN by allowing only recognized and managed services on those hosts to be accessible by hosts on the Internet. This kind of firewall architecture will be the one we will use throughout this book for all networking services and firewall implementations we want to install. A demilitarized zone (DMZ) is the most commonly used method in firewall security. All web, mail, ftp, lists, virtual, dns, database, and backup servers must be located in this zone. The gateway server also needs to be located in this zone since it makes the bridge between the private internal zone and the Internet.



The boxes between Hub A and B are in the 'DMZ'. Hub A only routes traffic between the Internet and the DMZ. Hub B only routes traffic between the DMZ and the Intranet. The theory is that all traffic between the Intranet and the Internet has to pass through a machine in the DMZ. The machine in the DMZ can be used to authenticate, record, and control all traffic via a Packet Filter Firewall or an Application Gateway Firewall software.

Linux `IPTables` Firewall Packet Filter

The new Linux kernel, like the two previous kernels, supports a new mechanism for building firewalls, called network packet filtering (netfilter). This new mechanism, which is controlled by a tool named `IPTables`, is more sophisticated than the previous `IPCHAINS` and more secure. This easy to configure new mechanism is also the first stateful firewall on a Linux operating system. Stateful firewalling represents a major technological jump in the intelligence of a firewall and allows administrators, for example, to block/detect many stealth scans that were undetected on previous generations of Linux firewalls. It also blocks most of the DoS attacks by rating limiting user-defined packet types, since it keeps each connection passing through it in memory.

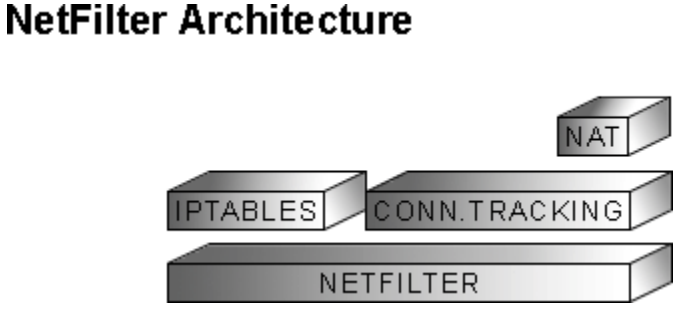
This new technology means that if a foreign packet tries to enter the network by claiming to be part of an existing connection, `IPTables` can consult its list of connections, which it keeps in memory, and if it finds that the packet doesn't match any of these, it will drop the packet which will defeat the scan in many cases! I would say that 50% of security on a network depends on a good firewall, and everyone should now be running at least `IPTables` on a Linux server to reach this level of security.

The Netfilter Architecture

Netfilter is used to forward, redirect, masquerade and filter packets coming into or out of our network. It is the framework of `IPTables` and without it `IPTables` will never work. Currently, four major subsystems exist on top of netfilter but only three are really important to make `IPTables` work. These three subsystems are what we regularly use to construct and build the rules and chains the firewall will use based on our policies.

These subsystems are:

- ✓ The '`iptables`' packet classification system.
- ✓ The connection-tracking system.
- ✓ The `NAT` system.



The 'iptables' packet classification system:

The IPTables packet classification system provides the "filter table" used by the packet filtering system to filter traffic, protocols and IP packets on the server. It is one of the most important subsystem components of IPTables. It works at the network level. Data is only allowed to leave the system if the firewall rules allow it. As packets arrive they are filtered by their type, source address, destination address, and port information contained in each packet. This is possible with the IPTables filter commands, which allow us to build rules to accomplish it. When we use and mix the IPTables filter commands together in the same line (because we know what each command means) we create a rule that the IPTables software understands and applies.

Many commands exist and it is not to our intention to list all of them here and explain each of them. We'll only show you the most important ones and their meanings. If you need more detailed information about each IPTables command and how to use them, please read a good firewall book or see the Netfilter web page. After reading this brief introductory chapter about IPTables, you should be able to understand the most important commands, e.g. how a rule is defined, as well as all the subsystem mechanisms of IPTables. This is all we need for the next chapter, where we'll install and configure the firewall software to interact with IPTables.

Now lets explain the most important parts of IPTables NetFilter.

The IPTables rules

Rules are used in IPTables to define what we want to do with the IP packets and protocols coming in to or out of our machine.

1. Each rule should be defined on one line for the firewall to separate rules.

2. Each new rule should begin with the word "iptables" which refers to the IPTables binary program that will be run.

The IPTables chains

Three built-in chains (INPUT, OUTPUT, and FORWARD) exist by default with IPTables. These are used to decide if the rule should be applied for INPUT packets, OUTPUT packets or FORWARDed packets. There are several ways to manipulate rules inside a chain.

1. We can append a new rule to a chain (-A).

2. Define which protocol to use (-p) to the chain.

3. Specifying the source (-s) and destination (-d) IP addresses to the chain.

4. Specifying the source (--sport) and destination (--dport) port range specification.

5. On which interface (-i for incoming packet on the interface and -o for outgoing packet on the interface) to match the rule, and so on.

The IPTables example for rules and chains

Really, we need to show an example to clarify the above. Imagine that we want to block any HTTP packets that come in or out of our server on the external network interface. Here are the rules to accomplish it.

The first rule (the complete line), instructs IPTables to add to its INPUT chain (-A INPUT) a new definition that will drop all packets (-j DROP) entering in the eth0 interface (-i eth0) using the TCP protocol (-p tcp) coming from anywhere (-s 0.0.0.0) on source port between 1024 & 65535 (--sport 1024:65535) to destination IP address 207.35.78.2 (-d 207.35.78.2) on the destination port 80 (--dport 80) for the HTTP service.

```
/sbin/iptables -A INPUT -i eth0 -p tcp -s 0.0.0.0 --sport 1024:65535 -d
207.35.78.2 --dport 80 -j DROP
```

The second rule, instructs IPTables to add to its OUTPUT chain (-A OUTPUT) a new definition that will drop all packets (-j DROP) going out on the eth0 interface (-i eth0) using the TCP protocol (-p tcp) coming from IP address 207.35.78.2 (-s 207.35.78.2) on source port 80 (--sport 80) to anywhere (-d 0.0.0.0) on destination ports between 1024 & 65535 (--dport 1024:65535) for the HTTP service.

```
/sbin/iptables -A OUTPUT -o eth0 -p tcp -s 207.35.78.2 --sport 80 -d 0.0.0.0 --
dport 1024:65535 -j DROP
```

In the above example, we have defined two new rules. The first rule is for incoming connections with the INPUT chain, and the second rule for outgoing connections with the OUTPUT chain.

The connection-tracking system:

It is with its "Connection Tracking" feature IPTables can recognize instructions to allow 'NEW' connections, 'RELATED' connections, etc. The connection-tracking subsystem is one of the pieces that makes IPTables more intelligent than its predecessor IPCHAINS. Connection Tracking keeps track of the relationships of packets by maintaining state information about a connection using memory tables. As mentioned previously, firewalls that do this are known as stateful. It is used when you add and declare "state" options like 'NEW', 'ESTABLISHED', 'RELATED', and 'INVALID' into your IPTables rules.

This feature becomes enabled when you define the "--state" option in your rules. The "state" feature gives you the opportunity to decide how incoming or outgoing connections should be analyzed and treated. To achieve this, the IPTables "state" feature provide us four possibilities.

1. NEW
 Allow an incoming or outgoing packet, which creates a new connection.

2. ESTABLISHED
 Allow an incoming or outgoing packet, which belongs to an existing connection.

3. RELATED
 Allow an incoming or outgoing packet, which is related to, but no part of, an existing connection.

4. INVALID
 Allow an incoming or outgoing packet, which could not be identified for some reason.

By using the above options with IPTables (highly recommended) we can fine tune our firewall and control much more tightly how packets should be treated before coming into or going out of our server.

The `IPTables` example for connection-tracking

Below are examples on how to use these options with `IPTables`. We use the previous rules for our example and then we add the connection-tracking feature to it. One other difference with the previous example is that instead of denying traffic we allow it to pass.

The first rule, instructs `IPTables` to add to its `INPUT` chain (`-A INPUT`) a new definition that will accept all packets, (`-j ACCEPT`) which may create new connections or they might belong to an existing connection (`-m state --state NEW,ESTABLISHED`), to enter in the `eth0` interface (`-i eth0`) with the `TCP` protocol (`-p tcp`) coming in from anywhere (`-s 0.0.0.0`) on source ports between 1024 & 65535 (`--sport 1024:65535`) to destination IP address `207.35.78.2` (`-d 207.35.78.2`) on destination port 80 (`--dport 80`) for the `HTTP` service.

```
/sbin/iptables -A INPUT –i eth0 -p tcp -s 0.0.0.0 --sport 1024:65535 -d
207.35.78.2 --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT
```

The second rule, instructs `IPTables` to add to its `OUTPUT` chain (`-A OUTPUT`) a new definition that will accept all packets (`-j ACCEPT`) which belong to an existing connection (`-m state --state ESTABLISHED`) to go out on the `eth0` interface (`-i eth0`) using the `TCP` protocol (`-p tcp`) coming from IP address `207.35.78.2` (`-s 207.35.78.2`) on source port 80 (`--sport 80`) to anywhere (`-d 0.0.0.0`) on destination ports between 1024 & 65535 (`--dport 1024:65535`) for `HTTP` service.

```
/sbin/iptables -A OUTPUT –o eth0 -p tcp -s 207.35.78.2 --sport 80 -d 0.0.0.0 --
dport 1024:65535 -m state --state ESTABLISHED -j ACCEPT
```

In the above example, we have been using two connection-tracking options to build the rules. For incoming connections, we use the "`NEW`" and "`ESTABLISHED`" options to inform `IPTables` to accept packets which create a new connection, and packets which belong to an existing connection. For outgoing connections, we only use "`ESTABLISHED`" to inform `IPTables` to accept packets, which belong to an existing connection.

The `INVALID` state should never be used, since its means that the packet is associated with no known connection. The `RELATED` state is used in some cases, for example, `FTP` data transfer or `ICPM` errors and means that the packet is starting a new connection, but is associated with an existing connection.

The NAT system:

NAT (**N**etwork **A**ddress **T**ranslation) transparently makes one set of IP addresses appear to be another set to the external world. It is used when you want to map an entire network onto one IP address, or when you want to forward certain connections to specific servers with private addresses or when you want to use load balancing to distribute load over several systems.

NAT can be divided into two different types: **S**ource **NAT** (SNAT) and **D**estination **NAT** (DNAT).

1. **S**ource **NAT** is when you alter the source address of the first packet (i.e. you are changing where the connection is coming from). **S**ource NAT is always done post-routing, just before the packet goes out onto the wire. Masquerading is a specialized form of SNAT, because you change the source address of the first packet.

2. **D**estination **NAT** is when you alter the destination address of the first packet (i.e. you are changing where the connection is going to). **D**estination **NAT** is always done pre-routing, when the packet first comes off the wire. Port forwarding, load sharing, and transparent proxying are all forms of DNAT, because you want people to be able to get to the boxes behind the one with the 'real' IP address.

The IPTables example for NAT

Below are examples on how to use these tables with IPTables. The table of NAT rules contains two lists called 'chains'. The two chains are PREROUTING (for **D**estination NAT, as packets first come in), and POSTROUTING (for **S**ource NAT, as packets leave).

For all the NAT operations that you want to do in your firewall script file, you will have to use the '-t nat' option to enable the NAT table feature of IPTables, since without this option, the NAT table will not work.

If you simply want to tell your Gateway Server that all packets coming from your internal network should be made to look like they are coming from the external interface (eth0) or from your dialup box (ppp0) then you would use the following rules:

```
/sbin/iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

This says the following to the system: In the NAT table (-t nat), append a rule (-A) after routing (POSTROUTING) for all packets going out eth0 (-o eth0), MASQUERADE the connection (-j MASQUERADE).

Now if you want to do port forwarding, meaning for example, that you want TCP packets coming into your external interface, which is directly connected to the Internet on IP address 207.35.78.2 port 8080, to have their destination mapped to your internal interface on IP address 192.168.1.1 on port 80, then you would use the following rules to achieve it.

```
/sbin/iptables -A PREROUTING -t nat -p tcp -d 1.2.3.4 --dport 8080 \
        -j DNAT --to 192.168.1.1:80
```

This says : Append a pre-routing rule (-A PREROUTING) to the NAT table (-t nat) so that TCP packets (-p tcp) going to 207.35.78.2 (-d 207.35.78.2) on port 8080 (--dport 8080) have their destination mapped (-j DNAT) to 192.168.1.1 on port 80 (--to 192.168.1.1:80).

As we can see, there are many options, parameters, and tables and it is very easy to make a mistake even if we are familiar with Firewall NetFilter technologies like `IPTables`. We can easily forget some important rule or even open some dangerous ports in error. Building a complete set of rules and chains suitable for all possible types of servers and workstations is a long task and it becomes evident that some predefined firewall rules are required to help us.

Conclusion

As a change to the previous books where we provided predefined firewall rules to include in your firewall script, we will use a different approach in this new edition of Securing & Optimizing Linux. Two mains reasons justify this change.

Firstly, Adrian Pascalau <`apascalau@openna.com`> has developed a new, very powerful and easy to use firewall software, based on my initial firewall work, which includes support for all possible requirements and needs of an `IPTables` firewall set-up and secondly because the previous predefined rules were not relevant to the needs of all users. If we had to cover every possible configuration, it would take a complete book in itself, therefore the best solution was to start a new piece in parallel and write a new firewall program based on `IPTables` that handles and covers, as much as possible, the needs of all Linux users. This has been done and I would like to thanks Adrian for his invaluable help, hard work and expertise in this area. `GIPTables-Firewall` is the result of this vision and it is the firewall program that we'll use and explain in the next chapter.

CHAPTER 10

`GIPTables` Firewall

IN THIS CHAPTER

1. **Building a kernel with `IPTables` support**
2. **Compiling - Optimizing & Installing `GIPTables`**
3. **Configuring `GIPTables`**
4. **`/etc/giptables.conf`: The `GIPTables` Configuration File**
5. **`/etc/rc.d/rc.giptables.blocked`: The `GIPTables` Blocked File**
6. **`/etc/init.d/giptables`: The `GIPTables` Initialization File**
7. **The `GIPTables` Firewall Module Files**
8. **How `GIPTables` parameters work?**
9. **Running the type of `GIPTables` firewall that you need**
10. **The `GIPTables` configuration file for a Gateway/Proxy Server**
11. **`GIPTables` Administrative Tools**

Linux GIPTables

Abstract

GIPTables Firewall is a free set of shell scripts that helps you generate Net filter/IPTables rules for Linux 2.4.x and newer kernels. It is very easy to configure and at present, designed to run on hosts with one or two network cards. It doesn't require that you to install any additional components to make it work with your Linux system. All you need to set-up a very secure firewall for your Linux machines is IPTables and GIPTables.

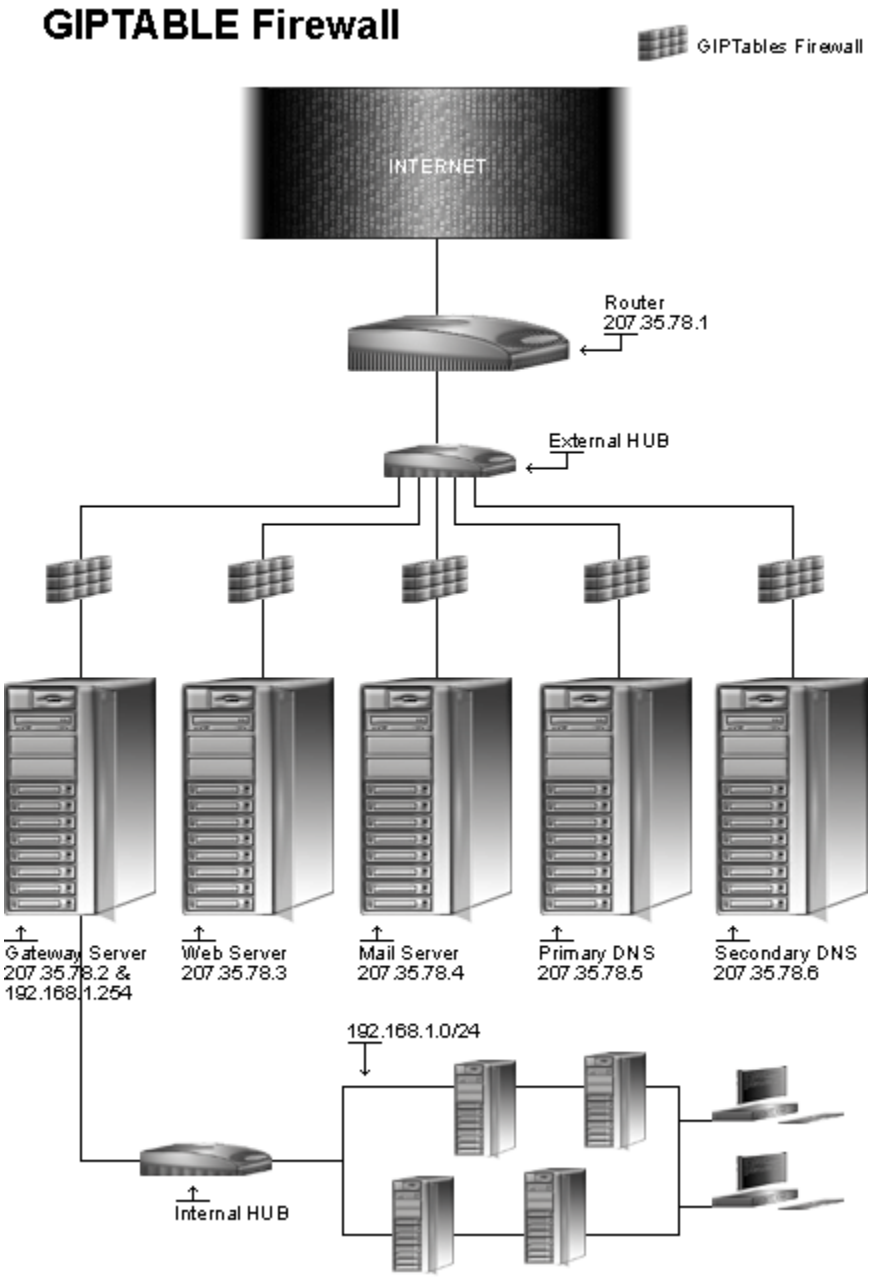
GIPTables can be used very easily with a host that has only one network card, and this host can be a server or a workstation. It assumes that if your host has two network cards, then the host should be a Gateway Server that connects your INTERNAL private network to the EXTERNAL world (the Internet).

Access from your internal network to the external world is automatically controlled and filtered by the SNAT feature of IPTables and GIPTables. This is well known in the Linux world as MASQUERADING. The DNAT feature of IPTables and GIPTables automatically controls access from the Internet to your internal servers where the software will forwards specified incoming connections to your internal server.

GIPTables-Firewall has many advantage compared to its competitors.

- ✓ It's easy to install and configure.
- ✓ It does not require you to install any additional component to make it work.
- ✓ It only needs IPTables to run.
- ✓ It's uses NAT & MASQ for sharing Internet access when you don't have enough IP.
- ✓ It's uses the stateful packet filtering (connection tracking) feature of IPTables.
- ✓ It's automatically does all kinds of network address translation.
- ✓ It's uses rate-limited connection and logging capability.
- ✓ It provides good protection against all kind of TCP SYN-flooding Denial of Service attacks.
- ✓ It provides good prootections against IP spoofing.
- ✓ It provides TCP packets heath check.
- ✓ It runs on any type of Linux system.
- ✓ It has a flexible and extensible infrastructure.
- ✓ It's easy to adjust and modify for your needs.
- ✓ It's small and does not use a lot of memory.
- ✓ It merges cleanly with all native Linux programs.
- ✓ It's well written and very powerful.
- ✓ It covers all needs in a highly secure server environment.

GIPTables-Firewall is simply the best firewall software to use with IPTables. It comes with a myriad ready to use of predefined rules. To be protected all we need to do is to answer in its configuration file 'Yes' or 'No' to the questions. Nothing more than that is required from your part to make it work.


GIPTABLE Firewall
GIPTables Firewall
INTERNET
Router
207.35.78.1
External HUB
Gateway Server
207.35.78.2 &
192.168.1.254
Web Server
207.35.78.3
Mail Server
207.35.78.4
Primary DNS
207.35.78.5
Secondary DNS
207.35.78.6
192.168.1.0/24
Internal HUB

These installation instructions assume

Commands are Unix-compatible.
The source path is `/var/tmp` (note that other paths are possible).
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation require using the super-user account "`root`".
Whether kernel recompilation may be required: Yes
Latest `GIPTables` version number is `1.1`

We have only tested `GIPTables` on OpenNA Linux and Red Hat Linux, but the procedures given in this chapter are likely to work on all Linux platforms.

Packages

The following is based on information as listed by `GIPTables-Firewall` as of 2002/06/09. Please regularly check at http://www.giptables.org/ for the latest status. We chose to install from source file because it provides us the opportunity to fine tune the installation.

Source code is available from:
`GIPTables-Firewall` Homepage: www.giptables.org
You must be sure to download: `giptables-1.1.tar.gz`

Prerequisites

Linux `GIPTables` requires that the listed software below is already installed on your system to be able to run and work successfully. If this is not the case, you must install them from your Linux CD-ROM or source archive file. Please make sure you have all of these programs installed on your machine before you proceed with this chapter.

- ✓ `Kernel 2.4` is required to set up `GIPTables` in your system.
- ✓ `iptables` package, is the new secure and more powerful program used by Linux to set up `GIPTables` in your system.

Building a kernel with `IPTables` support

The first thing you need to do is to ensure that your kernel has been built with the NetFilter infrastructure compiled in it: NetFilter is a general framework inside the Linux kernel, which other things (such as the `iptables` module) can plug into. This means you need kernel 2.4.x and answer "**`y`**", "**`n`**" or "**`m`**" to the following questions depending of the kernel type you have configured.

For a Monolithic Kernel, you would answer the questions "`y`" and your happier running a Modularized Kernel, you would answer the questions "`m`". It is important to understand that if `IPTables` is not enabled in your Kernel, NONE of the information contained in this chapter will work.

If your Kernel is one that comes directly from your Linux vendor or is unmodified, then there is a good chance that your kernel is already built to handle `IPTables`, therefore you wouldn't have to recompile it and/or go through the setup steps below.

Here are the required kernel setups for all type of servers except for a Gateway/Proxy:

*** Networking options**
*
Packet socket (CONFIG_PACKET) ← **Answer Y here**
Packet socket: mmapped IO (CONFIG_PACKET_MMAP) ← **Answer Y here**
Netlink device emulation (CONFIG_NETLINK_DEV) ← **Answer Y here**
Network packet filtering (replaces ipchains) (CONFIG_NETFILTER) ← **Answer Y here**
Network packet filtering debugging (CONFIG_NETFILTER_DEBUG) ← **Answer Y here**
Socket Filtering (CONFIG_FILTER) ← **Answer N here**
Unix domain sockets (CONFIG_UNIX) ← **Answer Y here**
TCP/IP networking (CONFIG_INET) ← **Answer Y here**
IP: multicasting (CONFIG_IP_MULTICAST) ← **Answer N here**
IP: advanced router (CONFIG_IP_ADVANCED_ROUTER) ← **Answer N here**
IP: kernel level autoconfiguration (CONFIG_IP_PNP) ← **Answer N here**
IP: tunneling (CONFIG_NET_IPIP) ← **Answer N here**
IP: GRE tunnels over IP (CONFIG_NET_IPGRE) ← **Answer N here**
IP: TCP Explicit Congestion Notification support (CONFIG_INET_ECN) ← **Answer N here**
IP: TCP syncookie support (disabled per default) (CONFIG_SYN_COOKIES) ← **Answer Y here**
*
*** IP: Netfilter Configuration**
*
Connection tracking (required for masq/NAT) (CONFIG_IP_NF_CONNTRACK) ← **Answer Y here**
FTP protocol support (CONFIG_IP_NF_FTP) ← **Answer Y here**
IRC protocol support (CONFIG_IP_NF_IRC) ← **Answer N here**
IP tables support (required for filtering/masq/NAT) (CONFIG_IP_NF_IPTABLES) ← **Answer Y here**
limit match support (CONFIG_IP_NF_MATCH_LIMIT) ← **Answer Y here**
MAC address match support (CONFIG_IP_NF_MATCH_MAC) ← **Answer Y here**
netfilter MARK match support (CONFIG_IP_NF_MATCH_MARK) ← **Answer Y here**
Multiple port match support (CONFIG_IP_NF_MATCH_MULTIPORT) ← **Answer Y here**
TOS match support (CONFIG_IP_NF_MATCH_TOS) ← **Answer Y here**
LENGTH match support (CONFIG_IP_NF_MATCH_LENGTH) ← **Answer Y here**
TTL match support (CONFIG_IP_NF_MATCH_TTL) ← **Answer Y here**
tcpmss match support (CONFIG_IP_NF_MATCH_TCPMSS) ← **Answer Y here**
Connection state match support (CONFIG_IP_NF_MATCH_STATE) ← **Answer Y here**
Packet filtering (CONFIG_IP_NF_FILTER) ← **Answer Y here**
REJECT target support (CONFIG_IP_NF_TARGET_REJECT) ← **Answer Y here**
Full NAT (CONFIG_IP_NF_NAT) ← **Answer N here**
Packet mangling (CONFIG_IP_NF_MANGLE) ← **Answer Y here**
TOS target support (CONFIG_IP_NF_TARGET_TOS) ← **Answer Y here**
MARK target support (CONFIG_IP_NF_TARGET_MARK) ← **Answer Y here**
LOG target support (CONFIG_IP_NF_TARGET_LOG) ← **Answer Y here**
TCPMSS target support (CONFIG_IP_NF_TARGET_TCPMSS) ← **Answer Y here**

Here are the required kernel setups for a Gateway/Proxy server:

*** Networking options**
*
Packet socket (CONFIG_PACKET) ← **Answer Y here**
Packet socket: mmapped IO (CONFIG_PACKET_MMAP) ← **Answer Y here**
Netlink device emulation (CONFIG_NETLINK_DEV) ← **Answer Y here**
Network packet filtering (replaces ipchains) (CONFIG_NETFILTER) ← **Answer Y here**
Network packet filtering debugging (CONFIG_NETFILTER_DEBUG) ← **Answer Y here**
Socket Filtering (CONFIG_FILTER) ← **Answer Y here**
Unix domain sockets (CONFIG_UNIX) ← **Answer Y here**
TCP/IP networking (CONFIG_INET) ← **Answer Y here**
IP: multicasting (CONFIG_IP_MULTICAST) ← **Answer Y here**
IP: advanced router (CONFIG_IP_ADVANCED_ROUTER) ← **Answer Y here**
IP: policy routing (CONFIG_IP_MULTIPLE_TABLES) ← **Answer Y here**

IP: use netfilter MARK value as routing key (CONFIG_IP_ROUTE_FWMARK) ← **Answer Y here**
IP: fast network address translation (CONFIG_IP_ROUTE_NAT) ← **Answer Y here**
IP: equal cost multipath (CONFIG_IP_ROUTE_MULTIPATH) ← **Answer Y here**
IP: use TOS value as routing key (CONFIG_IP_ROUTE_TOS) ← **Answer Y here**
IP: verbose route monitoring (CONFIG_IP_ROUTE_VERBOSE) ← **Answer Y here**
IP: large routing tables (CONFIG_IP_ROUTE_LARGE_TABLES) ← **Answer Y here**
IP: kernel level autoconfiguration (CONFIG_IP_PNP) ← **Answer N here**
IP: tunneling (CONFIG_NET_IPIP) ← **Answer Y here**
IP: GRE tunnels over IP (CONFIG_NET_IPGRE) ← **Answer Y here**
IP: TCP Explicit Congestion Notification support (CONFIG_INET_ECN) ← **Answer N here**
IP: TCP syncookie support (disabled per default) (CONFIG_SYN_COOKIES) ← **Answer Y here**
*
* **IP: Netfilter Configuration**
*
Connection tracking (required for masq/NAT) (CONFIG_IP_NF_CONNTRACK) ← **Answer Y here**
FTP protocol support (CONFIG_IP_NF_FTP) ← **Answer Y here**
IRC protocol support (CONFIG_IP_NF_IRC) ← **Answer Y here**
IP tables support (required for filtering/masq/NAT) (CONFIG_IP_NF_IPTABLES) ← **Answer Y here**
limit match support (CONFIG_IP_NF_MATCH_LIMIT) ← **Answer Y here**
MAC address match support (CONFIG_IP_NF_MATCH_MAC) ← **Answer Y here**
netfilter MARK match support (CONFIG_IP_NF_MATCH_MARK) ← **Answer Y here**
Multiple port match support (CONFIG_IP_NF_MATCH_MULTIPORT) ← **Answer Y here**
TOS match support (CONFIG_IP_NF_MATCH_TOS) ← **Answer Y here**
LENGTH match support (CONFIG_IP_NF_MATCH_LENGTH) ← **Answer Y here**
TTL match support (CONFIG_IP_NF_MATCH_TTL) ← **Answer Y here**
tcpmss match support (CONFIG_IP_NF_MATCH_TCPMSS) ← **Answer Y here**
Connection state match support (CONFIG_IP_NF_MATCH_STATE) ← **Answer Y here**
Packet filtering (CONFIG_IP_NF_FILTER) ← **Answer Y here**
REJECT target support (CONFIG_IP_NF_TARGET_REJECT) ← **Answer Y here**
Full NAT (CONFIG_IP_NF_NAT) ← **Answer Y here**
MASQUERADE target support (CONFIG_IP_NF_TARGET_MASQUERADE) ← **Answer Y here**
REDIRECT target support (CONFIG_IP_NF_TARGET_REDIRECT) ← **Answer Y here**
Packet mangling (CONFIG_IP_NF_MANGLE) ← **Answer Y here**
TOS target support (CONFIG_IP_NF_TARGET_TOS) ← **Answer Y here**
MARK target support (CONFIG_IP_NF_TARGET_MARK) ← **Answer Y here**
LOG target support (CONFIG_IP_NF_TARGET_LOG) ← **Answer Y here**
TCPMSS target support (CONFIG_IP_NF_TARGET_TCPMSS) ← **Answer Y here**
ipchains (2.2-style) support (CONFIG_IP_NF_COMPAT_IPCHAINS) ← **Answer N here**
ipfwadm (2.0-style) support (CONFIG_IP_NF_COMPAT_IPFWADM) ← **Answer N here**

WARNING: If you have followed the Linux Kernel chapter and have recompiled your Kernel, all the required options for `IPTables` firewall support, as shown above, are already set. Remember, all servers should be configured to block unused ports, even if they are not a firewall server.

Pristine source

As we don't use the RPM package to install the program, it would be difficult for us to locate all the files installed on the system if in the future we want to upgrade. To solve this problem, it is a good idea to make a list of files on the system before you install GIPTables, and then one afterwards, we can then compare them using the **diff** utility to find out what files were installed and where they were placed.

- Simply run the following command before installing the software:
  ```
  [root@deep root]# find /* > GIPTables1
  ```
- And the following one after you install the software:
  ```
  [root@deep root]# find /* > GIPTables2
  ```
- Then use the following command to get a list of what changed:
  ```
  [root@deep root]# diff GIPTables1 GIPTables2 > GIPTables-Installed
  ```

With this procedure, if any future upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. In the above example, we use the /root directory of the system to store the generated list of files.

Compiling - Optimizing & Installing GIPTables

To install the GIPTables software on your system, just download the latest version of the software from http://www.giptables.org/ site, and then as user 'root' expand the archive under your /var/tmp directory.

- To accomplish this use the following commands:
  ```
  [root@deep /]# cp giptables-version.tar.gz /var/tmp/
  [root@deep /]# cd /var/tmp/
  [root@deep tmp]# tar xzpf giptables-version.tar.gz
  ```

Next, move into the newly created GIPTables source directory and perform the following steps to install the software for your system.

- To move into the newly created GIPTables source directory use the command:
  ```
  [root@deep tmp]# cd giptables-1.1/
  ```
- To install GIPTables enter the following command:
  ```
  [root@deep giptables-1.1]# ./install.sh
  ```

The "install.sh" script file will simply install any GIPTables components on your system to the right location.

Once the installation of GIPTables has been completed, we can free up some disk space by deleting both the program tar archive and the related source directory since they are no longer needed.

- To delete GIPTables and its related source directory, use the commands:
  ```
  [root@deep /]# cd /var/tmp/
  [root@deep tmp]# rm -rf giptables-version/
  [root@deep tmp]# rm -f giptables-version.tar.gz
  ```

Configuring GIPTables

After GIPTables has been installed successfully on your system, your next step is to modify its configuration file to suit your needs. GIPTables is not software that needs to be compiled to work on your system but just to be configured. As you can imagine there are many possible configurations for a firewall design. Some may need to configure it to run for a Web Server, others may need to configure it to run a Mail Server, DNS Server, Virtual Server, Gateway Server, etc, or some others simply want to have the possibility to configure it for a specific requirement.

This is one of the advantages of GIPTables. It comes with many pre built configuration files which are suitable for many different types of server. The GIPTables configuration files are very flexible, easy to understand and setup. All you have to do is to answer questions, which refer to a specific firewall option either 'yes' to enable or 'no' to disable the service.

All pre built configuration files are located under the /lib/giptables/conf directory. Please, look in this directory for any existing configuration file relating to the version of the GIPTables software that you have. At the time writing, the following pre configured GIPTables configuration files are available.

giptables.conf.dns1	Default configuration file for a Master DNS Server.
giptables.conf.dns2	Default configuration file for a Slave DNS Server.
giptables.conf.ftpserver	Default configuration file for a FTP Server.
giptables.conf.gateway	Default configuration file for a Gateway Server.
giptables.conf.mailserver	Default configuration file for a Mail Server.
giptables.conf.ppp	Default configuration file for a dialup connection.
giptables.conf.README	Contains all possible configuration parameters.
giptables.conf.virtual	Default configuration file for a Virtual Server.
giptables.conf.webserver	Default configuration file for a Web Server.
giptables.conf.workstation	Default configuration file for a Workstation.

- ✓ /etc/giptables.conf (The GIPTables Configuration File)
- ✓ /etc/rc.d/rc.giptables.blocked (The GIPTables Blocked File)
- ✓ /etc/init.d/giptables (The GIPTables Initialization File)

/etc/giptables.conf: The GIPTables Configuration File

The /etc/giptables.conf file is the main configuration file for GIPTables. Though there are many options in this file, to get GIPTables up and running you should only need to change a small number of values.

But wait a minute, the giptables.conf file does not exist in /etc directory. Why? Remember that many possible firewall configurations exist and depending on both your requirements and the server type that you expect to protect, configurations may differ. GIPTables has some default example configuration files available under the /lib/giptables/conf directory that should suit your needs. You have to pick the one that is suitable for your server type and then create a symbolic link, as "giptables.conf" in the /etc directory that points to it. This is why the giptables.conf file doesn't exist in the /etc directory, it's purely a link.

Step1
First of all, choose one of the default configuration files that can be found in the /lib/giptables/conf directory. Find one that mostly meets your needs, make a backup copy of it and open it with any kind of text editor to configure it. In our example, we will configure our GIPTables firewall software for a Gateway/Proxy Server with two network interfaces since it is the most complicated configuration we are likely to encounter.

- These procedures can be accomplished with the following commands:
  ```
  [root@deep /]# cd /lib/giptables/conf/
  [root@deep conf]# cp giptables.conf.gateway giptables.conf.mybox
  [root@deep conf]# ln -sf /lib/giptables/conf/giptables.conf.mybox
  /etc/giptables.conf
  ```

In the above steps, we make a copy of our original "giptables.conf.gateway" file and create a symbolic link pointing to the copy.

> **NOTE:** It is a good idea not to directly modify an example configuration file, because if it gets damage, then you have to install the entire package again in order to get it back.

Step2
Now our giptables.conf file that is a symbolic link pointing to the original configuration file for a Gateway set-up exists. It is time to edit it and provide or change some minimal values to make it work for our system.

In the GIPTables configuration file below, we'll ONLY explain how to configure and set parameters that are the same for all types of GIPTables firewall configuration. Parts that differ are associated with different available GIPTables modules that must be loaded by the firewall configuration file to enable different services. All available modules with GIPTables firewall are explained later in this document.

- Edit the **giptables.conf** file (vi /etc/giptables.conf) and configure it to fit in with your system and networking setup. Text in bold is what you should change to make the firewall work with your server:

The Debug Definition:

The first configurable option that is the same for all types of GIPTables firewall configuration is "DEBUG" and by default it is set to "off". This option is useful only for debugging purposes.

```
# ------------------------------------------------------------------------
# DEBUG
#

   DEBUG="off"
```

If you set this option to "on", the firewall will display all IPTables rules relating to the GIPTables configuration file that you use to the screen, nothing will go to the kernel. The displayed set of rules will be commented so that you will not end up with lots of rules on the screen that you do not understand. This way you can see only what the firewall is generating, and also you will be able to better understand which rule is for what.

When this option is set to "off" (the default setting), the firewall will send all generated rules to the kernel, nothing will be displayed on your screen and the firewall will run on your system. Therefore is you want to run GIPTables on your system, you must be sure that this option is set to 'off'.

NOTE: When the "DEBUG" option is set to "on", it is possible to redirect the output of the firewall rules to a file, and use this file as learning example of how to set up IPTables rules for different kind of services. This is possible with the following command:

```
[root@deep tmp]# /etc/init.d/giptables start > output-result.txt
```

The Monolithic Kernel Definition:

The second configurable option that is the same for all types of GIPTables configuration is the "MONOLITIC_KERNEL" option and it is set to "no" by default. This option exists because the Linux kernel can be compiled so that it contains all the required IPTables driver code directly in it. This way we have a MONOLITHIC KERNEL and we would need to answer by "yes" to the option.

```
# ------------------------------------------------------------------------
# Some definitions for easy maintenance
# Edit these to suit your system
#

    MONOLITIC_KERNEL="no"
```

If you set this option to 'yes', then GIPTables will be informed that all native IPTables modules are directly compiled into the kernel. It is important to say 'yes' here only if you have a Monolithic Linux Kernel installed on your computer otherwise say 'no'. Then the firewall will look for and load all the IPTables modules that are required, depending on your configuration file.

NOTE: If you compile your kernel as Monolithic, you should know what IPTables modules you need to compile directly into the kernel, since the firewall will not try to load them. If you missed some modules, you will inevitably get errors, or the firewall might not work as expected. The best solution for a Monolithic Kernel set-up is to compile all native iptables modules into the kernel. Also, don't forget to set MONOLITIC_KERNEL="yes" in the firewall configuration file.

The External Network Interface Definition:

The next configurable option that is the same for all type of GIPTables firewall configuration is one of the most important settings in the configuration file.

```
# Interface 0: This is our external network interface
# It is directly connected to Internet

    INTERFACE0="eth0"
    INTERFACE0_IPADDR="x.x.x.x"
    ANY_IPADDR="0/0"
```

The above definitions set up the parameters associated with our network interface. The first parameter (**INTERFACE0="eth0"**) defines our external interface (the one directly connected to the Internet). By convention, we set it as 'eth0', but this is not mandatory and you can change it for whatever your external network interface is.

The second parameter (**INTERFACE0_IPADDR="x.x.x.x"**) defines the external IP address associated with the 'eth0' interface that your ISP or administrator assigned to you. Remember that every one will have a different IP address, therefore the IP value should be the one assigned to you.

The third parameter (**ANY_IPADDR="0/0"**) defines the IP address of any machine. The value of "0/0" means any machines from anywhere. This should NOT be changed, since we use this parameter when we want to talk to any machine out there.

WARNING: This warning apply only for a DHCP server configuration.

1) If you get your external IP address from your ISP's DHCP server, then set the value associated with the "INTERFACE0_IPADDR" parameter

To:

```
INTERFACE0_IPADDR=`/lib/giptables/if_ipaddr $INTERFACE0'.
```

2) Because the firewall is configured to be loaded before any network is initialized, we have to edit /etc/init.d/giptables file and replace the second line that reads:

```
# chkconfig: 2345 08 92
```

To read:

```
# chkconfig: 2345 11 92
```

Which will configure our firewall to start up after the network is initialized, and after we received our dynamic IP address from the DHCP server.

The Internal Network Interface Definition:

The next definition is very important for a ONLY Gateway Server set-up since it allows us to define our second network interface on the system. It is simply NOT required and does not apply on servers or workstations with only one network interface.

```
# Interface 1: This is our internal network interface
# It is directly connected to our internal Network 1

    INTERFACE1="eth1"
    INTERFACE1_IPADDR="192.168.1.254"
    NETWORK1="192.168.1.0/24"
```

The above definitions set up parameters associated with our second network interface (if any). As we can see, the first parameter (**INTERFACE1="eth1"**) defines, in this case, our internal interface name (the one directly connected to our internal private network).

The second parameter (**INTERFACE1_IPADDR="192.168.1.254"**) defines the internal IP address associated with the 'eth1' interface. Don't forget to change the IP address if your IP address is different.

Finally, the third and new parameter (**NETWORK1="192.168.1.0/24"**) defines our internal subnet. Note that we define it with the IP range to cover every node in our private internal network. As usual, you have to change the example IP address range for the one that you use.

> **NOTE:** If you do not have an internal network, then your machine is a Workstation or a Server with only one network interface. In this case just comment out those three options or only the INTERFACE1 option, and the firewall will totally ignore all other options that refer to the internal interface and network.
>
> If this is true in your case, then you will have to use another GIPTables example configuration file instead of the giptables.conf.gateway configuration file, which is only suitable for a Gateway Server.

The Name Servers Definition:

The Name Servers definition is where we define IP addresses for our Primary and Secondary Domain Name Servers. The entries can be the IP addresses that your ISP gave you or the one that your administrator gave you for your network.

```
# Your name servers ip address

    ISP_PRIMARY_DNS_SERVER="a.a.a.a"
    ISP_SECONDARY_DNS_SERVER="b.b.b.b"
```

The SYSLOG Server Definition:

The SYSLOG Server definition is mandatory. You only need to define it if you have one central log server configured to receive all syslog messages on your network. In general, and if you use this feature, you will have one server on your network configured to receive all log messages and all other servers configured to send their syslog message to the central log server. The value that should be entered into the SYSLOG Server Definition is the IP address of the central log server.

```
# SYSLOG server ip address

  SYSLOG_SERVER="c.c.c.c"
```

The loopback Interface Definition:

The next definition in our GIPTables configuration relates to the loopback interface of Linux and you don't need to modify it at all.

```
# Loopback interface

    LOOPBACK_INTERFACE="lo"                    # Loopback interface
```

The Ports Declarations Definition:

The same it true for the definition of privileged and unprivileged ports numbers on our system. The privileged ports numbers are used by daemon services that we run on the server and the unprivileged ports number by clients to establish a connection to the ports on the server. The ports declaration is important for `GIPTables` to distinguish which ports important services are allowed to run on and on which ports client are allowed to connect. This is a security feature.

```
# Port declarations do not change them

    PRIV_PORTS="0:1023"
    UNPRIV_PORTS="1024:65535"
```

The Custom Rules Definition:

Most services and rules used on production servers are already included with `GIPTables` through the different modules files. There can be situations where we need to add additional rules to the firewall; this is possible with the Custom Rules Definition. If you answer "`yes`" to the definition below, `GIPTables` will let you add you own custom `IPTables` rules through the file "`rc.giptables.custom`" located under the `/etc/rc.d` directory and the rules will then be added to the firewall.

```
# Loading custom firewall rules from /etc/rc.d/rc.giptables.custom
#

  LOAD_CUSTOM_RULES="yes"
```

If `LOAD_CUSTOM_RULES="no"`, then the Custom Rules Definition is disable.

The Logging Definition:

This section configures the logging of dropped packets and sends the logging information to a log file of our choice for later investigation. As you will see, for each interface and our internal network we have separate logging options. If you do not have an internal interface, then you can either just ignore, comment out or delete those options that refer to internal interface and internal network (Interface1 and Network1).

```
# --------------------------------------------------------------------------
# Logging

# We log & drop all the packets that are not expected. In order to avoid
# our logs being flooded, we rate limit the logging.

# Interface 0 log dropped packets

    INTERFACE0_LOG_DROPPED_PACKETS="yes"
    INTERFACE0_LOG_LIMIT="5/m"
    INTERFACE0_LOG_LIMIT_BURST="7"

# Interface 1 log dropped packets

    INTERFACE1_LOG_DROPPED_PACKETS="yes"
    INTERFACE1_LOG_LIMIT="7/m"
    INTERFACE1_LOG_LIMIT_BURST="9"

# Network 1 log forwarded dropped packets

    NETWORK1_LOG_DROPPED_PACKETS="yes"
    NETWORK1_LOG_LIMIT="9/m"
    NETWORK1_LOG_LIMIT_BURST="11"
```

Our default firewall policy is to DROP everything, and ACCEPT only wanted packets. In an ideal network environment, we do not need to drop a single packet, but when we want to protect our machine or our internal network from the garbage that is out there on the Internet then we really need to consider dropping unwanted packets.

What we actually drop are weird packets, incoming connections for services that we do not want to give to the external world, and so on. When those unwanted packets are coming in, we log them just to see when and from where those packets are coming in.

Now, there might be a situation when somebody out there will send to us only packets that we don't want, and because we are logging everything that we drop; soon our logs will fill our disk space. To avoid this, we impose a rate limit to the logging, so that at any time, only the value entered into the `LOG_LIMIT` parameter will be logged with a burst of the value entered into the `LOG_LIMIT_BURST` parameter.

The `LOG_LIMIT` module option specifies the maximum average number of matches to allow per second, minute, hour or day by using `/second` or `/s`, `/minute` or `/m`, `/hour` or `/h` and `/day` or `/d`.

The `LOG_LIMIT_BURST` module option specifies the exact number of packets to log picked up from the value defined in the `LOG_LIMIT` module option.

Ok, I'm pretty sure that this seems a little confusing.

Therefore, if we take the above `INTERFACE0` example, the definitions mean that, the first time this rule is reached, the packet will be logged; in fact, since the default burst is 7 (**`INTERFACE0_LOG_LIMIT_BURST="7"`**), the first seven packets will be logged. After this, it will be five minutes (**`INTERFACE0_LOG_LIMIT="5/m"`**) before a packet will be logged from this rule, regardless of how many packets reach it.

The log information is sent to the `/var/log/messages` file. There are different strings that can be used to interpret the `/var/log/messages` file in order to find different types of dropped packet information:

- ✓ `giptables-drop-src-ipaddr:`
 The packet was dropped based on the source IP address.

- ✓ `giptables-drop-dst-ipaddr:`
 The packet was dropped based on the destination IP address.

- ✓ `giptables-new-no-syn:`
 The TCP packet was dropped because it was a NEW one without SYN flag set.

- ✓ `giptables-fragments:`
 The packet was dropped because it was a fragment.

- ✓ `giptables-malformed-xmas:`
 The TCP packet was dropped because it looks like a malformed XMAS packet.

- ✓ `giptables-malformed-null:`
 The TCP packet was dropped because it looks like a malformed NULL packet.

The Network Ghouls Definition:

There might be situations when we would like to DROP connections to and from one or more IP addresses. This can be done using the Network Ghouls section and by changing the default value of 'no' to 'yes'.

```
# ------------------------------------------------------------------------
# Network Ghouls
# Refuse any connection from problem sites
#

    NETWORK_GHOULS="no"
```

To enable the Network Ghouls definition, we have to answer 'yes' to the first parameter (**NETWORK_GHOULS="yes"**). If (**NETWORK_GHOULS="no"**), this section is ignored by the firewall, and it doesn't matter how many IP addresses are added.

> **NOTE:** The list of IP addresses that will be blocked from having any kind of access to your server on all interfaces should be defined into the /etc/rc.d/rc.giptables.blocked file when the **NETWORK_GHOULS** parameter is set to "yes".

The Syn-flood Protection Definition:

To protect your machine from SYN-flooding **D**enial **o**f **S**ervice (DoS) attacks, the SYN_FLOOD_PROTECTION parameter should be set to 'yes'. This allows us to limit the number of incoming TCP connections, and at anytime have a well-defined number of allowed TCP connections on the system.

```
# ------------------------------------------------------------------------
# Syn-flood protection
# Limit the number of incoming tcp connections
#

    SYN_FLOOD_PROTECTION="yes"

# Interface 0 incoming syn-flood protection

    INTERFACE0_IN_SYN_FLOOD_PROTECTION="yes"
    INTERFACE0_IN_TCP_CONN_LIMIT="1/s"
    INTERFACE0_IN_TCP_CONN_LIMIT_BURST="3"

# Interface 1 incoming syn-flood protection

    INTERFACE1_IN_SYN_FLOOD_PROTECTION="yes"
    INTERFACE1_IN_TCP_CONN_LIMIT="3/s"
    INTERFACE1_IN_TCP_CONN_LIMIT_BURST="5"

# Network 1 forwarded incoming syn-flood protection

    NETWORK1_IN_SYN_FLOOD_PROTECTION="yes"
    NETWORK1_IN_TCP_CONN_LIMIT="5/s"
    NETWORK1_IN_TCP_CONN_LIMIT_BURST="7"
```

The TCP_CONN_LIMIT option specifies the maximum average number of new TCP packets that starts a new connection to be accepted per second, minute, hour or day by using /second or /s, /minute or /m, /hour or /h and /day or /d.

In our example, we have two interface definitions (**INTERFACE0** & **INTERFACE1**) and one network definition (**NETWORK1**). The network definition refers to our internal network and the SYN-flood protection feature is enabled on each one. If you don't have an internal interface, then just ignore the options that refer to internal interface and network (Interface1 and Network1).

If SYN_FLOOD_PROTECTION="no", then the entire SYN-flood protections section are ignored.

The Sanity Check Definition:

The SANITY_CHECK definition allows us to check the sanity (health) of packets that are coming in. If the (SANITY_CHECK) option is set to 'yes', Sanity Check protection with your firewall will be enabled.

```
# --------------------------------------------------------------------------
# Sanity check
#

    SANITY_CHECK="yes"

# Make sure NEW incoming TCP connections are SYN packets

    INTERFACE0_IN_DROP_NEW_WITHOUT_SYN="yes"
    INTERFACE1_IN_DROP_NEW_WITHOUT_SYN="yes"
    NETWORK1_IN_DROP_NEW_WITHOUT_SYN="yes"

# Drop all incoming fragments

    INTERFACE0_IN_DROP_ALL_FRAGMENTS="yes"
    INTERFACE1_IN_DROP_ALL_FRAGMENTS="yes"
    NETWORK1_IN_DROP_ALL_FRAGMENTS="yes"

# Drop all incoming malformed XMAS packets

    INTERFACE0_IN_DROP_XMAS_PACKETS="yes"
    INTERFACE1_IN_DROP_XMAS_PACKETS="yes"
    NETWORK1_IN_DROP_XMAS_PACKETS="yes"

# Drop all incoming malformed NULL packets

    INTERFACE0_IN_DROP_NULL_PACKETS="yes"
    INTERFACE1_IN_DROP_NULL_PACKETS="yes"
    NETWORK1_IN_DROP_NULL_PACKETS="yes"
```

There are 4 different kinds of sanity checks used in this version of GIPTables Firewall and each one has a specific function to accomplish, which are.

A) Make sure that NEW incoming TCP connections are SYN packets. This will log and drop any new packet that does not have SYN flag set.

B) Drop all incoming fragments. This will log and drop any fragment. Fragments can be overlapped, and the subsequent interpretation of such fragments is very OS-dependent. In our protection, we are not going to trust any fragments, thus we log them just to see if we get any, and drop them too.

C) Drop all incoming malformed XMAS packets. A typical XMAS scan will most likely show all flags from TCP packet header set. We log and drop all XMAS packets.

D) Drop all incoming malformed NULL packets. A NULL packet has no flags set in the TCP header, so it does not do anything and we don't need it. Those NULL packets are usually used for port scans; therefore we should safely drop all of them.

You can set the sanity check protection based on interface or network. If you don't have an internal interface, then just ignore, comment out or delete the options that refer to internal interface and network (Interface1 and Network1).

If `SANITY_CHECK="no"`, then the entire sanity check section is ignored.

The Spoofing & Bad Addresses Definition:

All IP packet headers contain the source and destination IP addresses and the type of IP protocol message (ICMP, UDP or TCP) the packet contains. The only means of identification under the Internet Protocol (IP) is the source address in the IP packet header. This is a problem that opens the door to source address spoofing, where the sender may replace its address with either a nonexistent address, or the address of some other site.

Also, there are at least seven sets of source addresses you should always refuse on your external interface.

These are incoming packets claiming to be from:

- ✓ Your external IP address
- ✓ Class A private IP addresses
- ✓ Class B private IP addresses
- ✓ Class C private IP addresses
- ✓ Class D multicast addresses
- ✓ Class E reserved addresses
- ✓ The loopback interface

With the exception of your own IP address, blocking outgoing packets containing these source addresses also protects you from possible configuration errors on your part.

In this section we log and drop all incoming packets with source IP addresses that we do not expect or want. There are some important one that really need to be monitored and controlled as shown below:

```
# ------------------------------------------------------------------------
# Spoofing and bad addresses
#

    REFUSE_SPOOFING="yes"
```

There is no way for a packet that come in from the Internet on our external interface to have its source IP address the same with our external IP address. If this happen, then packets are spoofed; therefore we log and drop them.

A) We log and drop all incoming packets claiming to be from the IP addresses of our interfaces. In a Gateway firewall configuration, we have two network interfaces, and two IP addresses associated with them. Therefore, we should protect both interfaces as follow.

```
# Refuse incoming packets claiming to be from the IP addresses of our
interfaces

   REFUSE_SPOOFING_IPADDR[0]=$INTERFACE0_IPADDR
   INTERFACE0_IN_REFUSE_SPOOFING[0]="yes"
   INTERFACE1_IN_REFUSE_SPOOFING[0]="no"
   NETWORK1_IN_REFUSE_SPOOFING[0]="yes"

   REFUSE_SPOOFING_IPADDR[1]=$INTERFACE1_IPADDR
   INTERFACE0_IN_REFUSE_SPOOFING[1]="no"
   INTERFACE1_IN_REFUSE_SPOOFING[1]="yes"
   NETWORK1_IN_REFUSE_SPOOFING[1]="no"
```

B) We log and drop all incoming packets claiming to be from the broadcast source address range. We accept broadcast source packets only in one situation: when we have a DHCP Server, and this, because a DHCP Client will request its IP address by sending out and DHCP discovery packet that has source IP address "0.0.0.0" and destination IP address "255.255.255.255". In this situation, the Gateway Server is also a DHCP Server, so we will accept by default those broadcast source packets only on the internal interface.

```
# Refuse incoming packets claiming to be from broadcast-src address range

    REFUSE_SPOOFING_IPADDR[2]="0.0.0.0/8"
    INTERFACE0_IN_REFUSE_SPOOFING[2]="yes"
    INTERFACE1_IN_REFUSE_SPOOFING[2]="no"
    NETWORK1_IN_REFUSE_SPOOFING[2]="yes"
```

C) We log and drop all incoming packets claiming to be from the reserved loopback IP address range. This is so obvious. We should never have incoming packets with source IP address from the loopback address range. We can refuse them safely on all our interfaces.

```
# Refuse incoming packets claiming to be from reserved loopback address range

    REFUSE_SPOOFING_IPADDR[3]="127.0.0.0/8"
    INTERFACE0_IN_REFUSE_SPOOFING[3]="yes"
    INTERFACE1_IN_REFUSE_SPOOFING[3]="yes"
    NETWORK1_IN_REFUSE_SPOOFING[3]="yes"
```

E) We log and drop all incoming packets claiming to be from the well-known private networks: A, B, C. We can safely refuse all packets claiming to be from those private networks on all of our interfaces, and internal network.

```
# Refuse incoming packets claiming to be from class A private network

    REFUSE_SPOOFING_IPADDR[4]="10.0.0.0/8"
    INTERFACE0_IN_REFUSE_SPOOFING[4]="yes"
    INTERFACE1_IN_REFUSE_SPOOFING[4]="yes"
    NETWORK1_IN_REFUSE_SPOOFING[4]="yes"
```

```
# Refuse incoming packets claiming to be from class B private network

    REFUSE_SPOOFING_IPADDR[5]="172.16.0.0/12"
    INTERFACE0_IN_REFUSE_SPOOFING[5]="yes"
    INTERFACE1_IN_REFUSE_SPOOFING[5]="yes"
    NETWORK1_IN_REFUSE_SPOOFING[5]="yes"

# Refuse incoming packets claiming to be from class C private network

    REFUSE_SPOOFING_IPADDR[6]="192.168.0.0/16"
    INTERFACE0_IN_REFUSE_SPOOFING[6]="yes"
    INTERFACE1_IN_REFUSE_SPOOFING[6]="no"
    NETWORK1_IN_REFUSE_SPOOFING[6]="yes"
```

WARNING: There is only one exception in which case we do not refuse incoming packets on our internal interface claiming to be from our internal private network. This appears only for a Gateway Server when your internal network is from class C. You should not refuse incoming packets on internal interface from your internal network.

F) We log and drop all incoming packets claiming to be from class D, E, and unallocated IP addresses. These are classes that are not currently used or that are unallocated. There is no reason for an incoming packet to have a source IP address from one of those classes.

```
# Refuse incoming packets claiming to be from class D, E, and unallocated

    REFUSE_SPOOFING_IPADDR[7]="224.0.0.0/3"
    INTERFACE0_IN_REFUSE_SPOOFING[7]="yes"
    INTERFACE1_IN_REFUSE_SPOOFING[7]="yes"
    NETWORK1_IN_REFUSE_SPOOFING[7]="yes"
```

The above Spoofing and bad address protection assume that you have two network interfaces installed on your system. This configuration is suitable for a Gateway Server. If you only have one network interface on your server, then you can ignore, comment out or remove those options that refer to internal interface and network (Interface1 and Network1).

If `REFUSE_SPOOFING="no"` then the entire spoofing protection section is ignored.

The above configuration closes our discussion about parameters that are the same for all types of `GIPTables` firewall configurations. Once you have configured all of the customized values in this part of the `GIPTables` configuration file, suitable for your type of system, you are ready to start the software.

`/etc/rc.d/rc.giptables.blocked`: The `GIPTables` Blocked File

Sometimes you'll know an address that you would like to block from having any access at all to your server. Instead of entering the entire `iptables` line per IP address for those jerks on the internet, you can write them into the `rc.giptables.blocked` file, that will take the IP addresses, strip out any comments and run the resulting list through an `iptables` routine.

The net effect is the `/etc/giptables.conf` file increases no more than needed, especially when one might have a large number of IP addresses to deny.

Step 1
Edit the **rc.giptables.blocked** file (`vi /etc/rc.d/rc.giptables.blocked`) and add all the IP addresses that you want blocked from having any access to your server. For example, I've put the following IP addresses in this file:

```
# ------------------------------------------------------------------------
# GIPTables Firewall v0.1-fox
# Copyright (C) 2001, 2002 Adrian Pascalau <apascalau@openna.com>
# rc.giptables.blocked file
#
# ------------------------------------------------------------------------
# This file is part of GIPTables Firewall
#
# GIPTables Firewall is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA

204.254.45.9    # Cracker site with priority 01.
187.231.11.5    # Spam site with priority 07.
#214.34.144.4   # Temporally reactivated, please verify with log file.

# ------------------------------------------------------------------------
# End of file
```

Here we can see how this file can be useful. Now we can add the bad IP address, with some comments if necessary to remember why we've added the IP address, into the `/etc/rc.d/rc.giptables.blocked` file and restart `GIPTables` for the changes to take effect.

`/etc/init.d/giptables`: The `GIPTables` Initialization File

The `/etc/init.d/giptables` script file is responsible for automatically starting and stopping the `GIPTables` Firewall. It can take several parameters like '`start`', '`stop`', '`restart`' and '`panic`'. The '`start`' parameter will actually start the firewall, read the configuration file, clear any pre-defined rules and chains from the kernel, set DROP as the default policy which will deny everything by default and then generate the `IPTables` rules according to your `GIPTables` configuration file.

The '`stop`' parameter will stop the firewall, clear any pre-defined rules and chains from the kernel, and set ACCEPT as the default policy for all `IPTables` default chains. The '`restart`' option is really just '`start`' as this firewall isn't a daemon and '`start`' clears any pre-defined rules anyway. This is really only here to make those who expect it happy.

The '`panic`' option should be used when you want to cut any connections to and from your machine. It will clear any pre-defined rules and chains from the kernel, set default policy as DROP for all `IPTables` default chains and let through only the packets destined for the loopback interface.

- To start GIPTables on your system, use the following command:
 [root@deep /]# **/etc/init.d/giptables start**

The GIPTables Firewall Module Files

Once you have chosen the right GIPTables configuration file suitable for the type of server that you want to protect and all the parameters, which are the same for all type of GIPTables firewall have been configured, GIPTables should have enough information to run properly on your system. Really, your work is done and your firewall is well protected against attacks.

Everyone has a different set-up for his firewall design and sometimes we need to implement a new service and then open and control the port associated with this service on our server. GIPTables allows us to add, modify, delete, and customize any existing or expected services in a simple manner through its modules feature.

With GIPTables, each service like DNS, FTP, HTTPD, etc have their own modules. Those modules are loaded only when defined in the giptables.conf file, so that if there are no options related to FTP for example, the FTP module will not be loaded. You can specify on which interface or network the module will work, and what kind of requests (incoming or outgoing) can go thought that interface or network.

All GIPTables modules are located under the /lib/giptables/modules directory and it's in these module files that we handle all rules relating to the specific service. When we configure, customize and enable service parameters in the giptables.conf file, the parameter in question get its information about IPTables rules that must be used through the modules files available under the /lib/giptables/modules directory. If the parameter of the specific service that you want to enable is not defined into the GIPTables configuration file, then this service will not load its IPTables rules from its modules file and will not run with your GIPTables Firewall software.

If you look in the /lib/giptables/modules directory, you'll find the following modules for the services that can be enabled with GIPTables Firewall.

giptables-ANY	The ANY module, which refer to ANY services
giptables-AUTH	The AUTH module, which refer to AUTH services
giptables-DHCP	The DHCP module, which refer to DHCP services
giptables-DNS	The DNS module, which refer to DNS services
giptables-FINGER	The FINGER module, which refer to FINGER services
giptables-FTP	The FTP module, which refer to FTP services
giptables-HTTP	The HTTP module, which refer to HTTP services
giptables-HTTPS	The HTTPS module, which refer to HTTPS services
giptables-ICMP	The ICMP module, which refer to ICMP services
giptables-IMAP	The IMAP module, which refer to IMAP services
giptables-IMAPS	The IMAPS module, which refer to IMAPS services
giptables-LDAP	The LDAP module, which refer to LDAP services
giptables-LDAPS	The LDAPS module, which refer to LDAPS services
giptables-MYSQL	The MYSQL module, which refer to MYSQL services
giptables-NETBIOS	The NetBIOS module, which refer to NetBIOS services
giptables-NNTP	The NNTP module, which refer to NNTP services
giptables-NNTPS	The NNTPS module, which refer to NNTPS services
giptables-NTP	The NTP module, which refer to NTP services
giptables-ORACLE	The ORACLE module, which refer to ORACLE services
giptables-POP3	The POP3 module, which refer to POP3 services
giptables-POP3S	The POP3S module, which refer to POP3S services
giptables-POSTGRES	The POSTGRES module, which refer to POSTGRES services

giptables-SMTP	The SMTP module, which refer to SMTP services
giptables-SMTPS	The SMTPS module, which refer to SMTPS services
giptables-SQUID	The SQUID module, which refer to SQUID services
giptables-SSH	The SSH module, which refer to SSH services
giptables-SYSLOG	The SYSLOG module, which refer to SYSLOG services
giptables-TELNET	The TELNET module, which refer to TELNET services
giptables-TELNETS	The TELNETS module, which refer to TELNETS services
giptables-TRACEROUTE	The TRACEROUTE module, which refer to TRACEROUTE services
giptables-WEBCACHE	The WEBCACHE module, which refer to WEBCACHE services
giptables-WHOIS	The WHOIS module, which refer to WHOIS services

How GIPTables parameters work?

As we've shown, GIPTables modules are ONLY loaded when we define and enable their parameters into the GIPTables configuration file. Therefore, if we want to add to our existing configuration a new service that doesn't exist, we have to define, enable and configure the service with the right parameters.

The best way to get an idea about the implementation is to include a new service into our existing GIPTables configuration file. In our next example, we will add the MySQL service to our Gateway Server GIPTables Firewall. We'll go through the steps that you need to do to add the MySQL service to your GIPTables Firewall. Note that all of the following steps will be the same for any additional services that you might want to add to your existing GIPTables configuration file.

Step1
The first step will be to enable the MySQL service module into the GIPTables configuration file. We do this by adding the following lines into the file. Text in bold is what should be added to enable the example MySQL service.

- Edit the **giptables.conf** file (vi /etc/giptables.conf) and add the line.

 ACCEPT_MYSQL="yes"

The above line informs the software to enable the MySQL module service for the MySQL database on any network interfaces or network present on the system and for any requests (incoming or outgoing).

Step2
Once the MySQL module service has been enabled, we need to add the right parameters lines specific to the MySQL service to the GIPTables configuration file. Remember that GIPTables is a flexible program that lets us control traffic on external interface, internal interface, and internal network for incoming and outgoing traffic. For a Gateway Server, all options are required but for a server with one network interface, we only need to control traffic on the external interface for incoming and outgoing packets.

> **NOTE:** It is important to note that each GIPTables parameter has the same definition and only parts, which relate to services that we want to define change.

Enabling outgoing client requests
In the example below, we define and enable MySQL outgoing client requests for a Gateway Server. The difference about parameters with other type of servers is that we need to define additional network interface (INTERFACE1) and network (NETWORK1) for a Gateway Server set-up. All text in bold should be configured to define and enable the MySQL service.

```
# ------------------------------------------------------------------------
# MYSQL outgoing client request
#

# Interface 0 MYSQL outgoing client request

    INTERFACE0_MYSQL_CLIENT="yes"

    INTERFACE0_MYSQL_OUT_SRC_IPADDR[0]=$INTERFACE0_IPADDR
    INTERFACE0_MYSQL_OUT_DST_IPADDR[0]=$ANY_IPADDR
```

In the above example, we first enable MySQL outgoing client request on the external interface (**INTERFACE0_MYSQL_CLIENT="yes"**).

Next, we instruct the system that the parameters apply to interface 0 (**INTERFACE0**) for the MySQL service (**MYSQL**) for outgoing requests (**OUT**) with the source IP address (**SRC_IPADDR**) coming from our external interface IP address (**$INTERFACE0_IPADDR**). Which means, packets having our external interface IP address, as a source IP address will be able to go out and/or start a new connection.

Finally, we inform the system that the parameters also apply to interface 0 (**INTERFACE0**) for the MySQL service (**MYSQL**) for outgoing requests (**OUT**) with the destination IP address (**DST_IPADDR**) going to anywhere (**$ANY_IPADDR**). And this means, packets having our external interface, as the destination IP address will be able to go out and/or start a new connection.

Using the connection tracking capability of IPTables, the related MySQL incoming packets are automatically allowed back in by the firewall. In this case, our machine can be a MySQL client that is allowed to access any MySQL server on the Internet.

If we want to restrict access to only one external MySQL server, the parameters should be configured like in the example below:

```
# Interface 0 MYSQL outgoing client request

    INTERFACE0_MYSQL_CLIENT="yes"

    INTERFACE0_MYSQL_OUT_SRC_IPADDR[0]=$INTERFACE0_IPADDR
    INTERFACE0_MYSQL_OUT_DST_IPADDR[0]="x.x.x.x"
```

In this case, "x.x.x.x" is the IP address of the external MySQL server that we want to access. For a second MySQL server, another set of parameters should be added, like in the example below:

```
# Interface 0 MYSQL outgoing client request

    INTERFACE0_MYSQL_CLIENT="yes"

    INTERFACE0_MYSQL_OUT_SRC_IPADDR[0]=$INTERFACE0_IPADDR
    INTERFACE0_MYSQL_OUT_DST_IPADDR[0]="x.x.x.x"

    INTERFACE0_MYSQL_OUT_SRC_IPADDR[1]=$INTERFACE0_IPADDR
    INTERFACE0_MYSQL_OUT_DST_IPADDR[1]="y.y.y.y"
```

"x.x.x.x" is the IP address of the first external MySQL server that we want to access and "y.y.y.y" is the IP address of the second external MySQL server that we want to access. Please note that the index of parameters has been increased, so that the first set of parameters have the index 0, and the second set of parameters have the index 1.

NOTE: This rule is the same for all GIPTables Firewall parameters that have an index. If you would like to add a second set of parameters, just copy/paste them, make the required changes and do not forget to increase the index.

On a Gateway Server or machines with two networks interfaces, we need to define the following additional parameters for the firewall to recognize the other network interface and the private network behind it.

```
# Interface 1 MYSQL outgoing client request

    INTERFACE1_MYSQL_CLIENT="yes"

    INTERFACE1_MYSQL_OUT_SRC_IPADDR[0]=$INTERFACE1_IPADDR
    INTERFACE1_MYSQL_OUT_DST_IPADDR[0]=$NETWORK1
```

In the above example, we enable MySQL outgoing client request on the internal interface (**INTERFACE1_MYSQL_CLIENT="yes"**).

We instruct the system that the parameters apply to internal interface 1 (**INTERFACE1**) for the MySQL service (**MYSQL**) to outgoing requests (**OUT**) with source IP address (**SRC_IPADDR**) coming from our internal interface IP address (**$INTERFACE1_IPADDR**). Therefore, any packets having our internal interface IP address, as source IP address will be able to go out and/or start a new connection.

Next, we inform the system that the parameters also apply to internal interface 1 (**INTERFACE1**) for the MySQL service (**MYSQL**) for outgoing requests (**OUT**) with a destination IP address (**DST_IPADDR**) going from our internal subnet IP address range (**$NETWORK1**). Therefore, any packets from our internal subnet will be able to go out and/or start new connections.

Using the connection tracking capability of IPTables, the related MySQL incoming packets are automatically allowed back in by the firewall. In this case, our machine can be a MySQL client that is allowed to access any MySQL server from our internal subnet.

```
# Network 1 MYSQL forwarded outgoing client request

    NETWORK1_MYSQL_CLIENT="yes"

    NETWORK1_MYSQL_OUT_SRC_IPADDR[0]=$NETWORK1
    NETWORK1_MYSQL_OUT_DST_IPADDR[0]=$ANY_IPADDR
```

Here, we enable MySQL outgoing client requests on our internal subnet (**NETWORK1_MYSQL_CLIENT="yes"**).

We instruct the system that the parameters apply to our internal subnet (**NETWORK1**) for the MySQL service (**MYSQL**) for outgoing requests (**OUT**) with the source IP address (**SRC_IPADDR**) coming from our internal subnet IP address range (**$NETWORK1**).

In the second line, we inform the system that the parameters also apply to our internal subnet (**NETWORK1**) for the MySQL service (**MYSQL**) to outgoing requests (**OUT**) with destination IP address (**DST_IPADDR**) going to anywhere (**$ANY_IPADDR**).

Using the connection tracking capability of IPTables, the related MySQL incoming packets are automatically allowed back in by the firewall. In this case, our machines from our internal subnet are the MySQL clients and are allowed to access any MySQL server on the Internet.

NOTE: The requests are automatically SNATed (MASQUERADEd) by the GIPTables Firewall, so that the MySQL server from the Internet thinks that talks with our Gateway server.

In general, you should only replace MYSQL with the name of the service that you want to define for the parameters to work for other type of services. In our example, we use MYSQL; it is to you to change it for the service of your choice.

Enabling incoming client requests

As we can see, all of the above parameters apply only to outgoing client requests for the MySQL service on a Gateway Server. Now for incoming server requests, we should add the related lines to the configuration file to allow them in. In the example below, we define and enable MySQL incoming client requests for a Gateway Server. The difference with parameters for other types of servers is that here we need to define an additional network interface (INTERFACE1) and a network (NETWORK1) for a Gateway Server set-up.

```
# ------------------------------------------------------------------------
# MYSQL incoming client request
#

# Interface 0 MYSQL incoming client request

    INTERFACE0_MYSQL_SERVER="yes"

    INTERFACE0_MYSQL_IN_SRC_IPADDR[0]=$ANY_IPADDR
    INTERFACE0_MYSQL_IN_DST_IPADDR[0]=$INTERFACE0_IPADDR
```

In the above example, we first enable incoming client request for MySQL on the external interface (**INTERFACE0_MYSQL_SERVER="yes"**).

Next, we instruct the system that the parameters apply to external interface 0 (**INTERFACE0**) for the MySQL service (**MYSQL**) for incoming requests (**IN**) with the source IP address (**SRC_IPADDR**) coming from anywhere (**$ANY_IPADDR**). This mean that we permit the firewall to receive packets coming from anywhere on our external interface to start a new connection.

Finally, we inform the system that parameters also apply to external interface 0 (**INTERFACE0**) for MySQL service (**MYSQL**) on incoming requests (**IN**) with destination IP address (**DST_IPADDR**) coming from our external IP address (**$INTERFACE0_IPADDR**). In other terms, incoming packets having our external interface, as destination IP address will be able to come in and/or start a new connection.

Using the connection tracking capability of IPTables, the related MySQL outgoing packets are automatically allowed back out by the firewall. In this case, our machine is a MySQL server that is allowed to receive requests from any MySQL client from the Internet.

If we want to allow access to only one external client machine on the MySQL server, the parameters should be configured like in the example below:

```
# Interface 0 MYSQL incoming client request

    INTERFACE0_MYSQL_SERVER="yes"

    INTERFACE0_MYSQL_IN_SRC_IPADDR[0]="x.x.x.x"
    INTERFACE0_MYSQL_IN_DST_IPADDR[0]=$INTERFACE0_IPADDR
```

In this case, "x.x.x.x" is the IP address of the external client machine that is allowed to access our MySQL server. For a second external client machine allowed, another set of parameters should be added, like in the example below:

```
# Interface 0 MYSQL incoming client request

    INTERFACE0_MYSQL_SERVER="yes"

    INTERFACE0_MYSQL_IN_SRC_IPADDR[0]="x.x.x.x"
    INTERFACE0_MYSQL_IN_DST_IPADDR[0]=$INTERFACE0_IPADDR

    INTERFACE0_MYSQL_IN_SRC_IPADDR[1]="y.y.y.y"
    INTERFACE0_MYSQL_IN_DST_IPADDR[1]=$INTERFACE0_IPADDR
```

"x.x.x.x" is the IP address of the first external client machine that is allowed to access our MySQL server and "y.y.y.y" is the IP address of the second external client machine that is allowed to access our MySQL server. Please note that the index of parameters has been increased, so that the first set of parameters have the index 0, and the second set of parameters have the index 1.

NOTE: This rule is the same for all GIPTables Firewall parameters that have an index. If you would like to add a second set of parameters, just copy/paste them, make the required changes and do not forget to increase the index.

Don't forget that we need to add all of the lines below for a Gateway Server set-up for the firewall to recognize the second network interface and our internal subnet. The definitions and explanations are the same as for outgoing client requests explained earlier.

```
# Interface 1 MYSQL incoming client request

    INTERFACE1_MYSQL_SERVER="yes"

    INTERFACE1_MYSQL_IN_SRC_IPADDR[0]=$NETWORK1
    INTERFACE1_MYSQL_IN_DST_IPADDR[0]=$INTERFACE1_IPADDR
```

In the above example, we enable MySQL incoming client request on the internal interface (**INTERFACE1_MYSQL_SERVER="yes"**).

Next, we instruct the firewall that on the internal interface (**INTERFACE1**), all MySQL (**MYSQL**) incoming packets (**IN**) with source IP address (**SRC_IPADDR**) from our internal subnet IP address range (**$NETWORK1**) and with destination IP address (**DST_IPADDR**) coming from our internal interface IP address (**$INTERFACE1_IPADDR**) will be allowed to come in and/or start a new connection.

In other terms, any incoming MySQL packets with source IP address from our internal subnet IP address range and with our internal interface IP address as destination IP address will be allowed to come in and/or start a new connection.

Using the connection tracking capability of IPTables, the related MySQL outgoing packets are automatically allowed back out by the firewall. In this case, our machine is a MySQL server that is allowed to receive requests from any MySQL client from our internal subnet.

There might be a situation when we would like to access the MySQL server from our internal subnet using the external interface IP address (**$INTERFACE0_IPADDR**) as destination IP address (**DST_IPADDR**). This is the case when we connect to the MySQL server using its host name instead of the IP address. Our DNS server might resolve the MySQL server's IP address as the external interface IP address. In this case, the parameters should be configured like in the example below:

```
# Interface 1 MYSQL incoming client request

    INTERFACE1_MYSQL_SERVER="yes"

    INTERFACE1_MYSQL_IN_SRC_IPADDR[0]=$NETWORK1
    INTERFACE1_MYSQL_IN_DST_IPADDR[0]=$INTERFACE1_IPADDR

    INTERFACE1_MYSQL_IN_SRC_IPADDR[1]=$NETWORK1
    INTERFACE1_MYSQL_IN_DST_IPADDR[1]=$INTERFACE0_IPADDR
```

As you can see, we have copy/paste the first set of parameters, then changes the destination IP address (**DST_IPADDR**) to our external interface IP address (**$INTERFACE0_IPADDR**) and also increase the index number.

```
# Network 1 MYSQL forwarded incoming server request

    NETWORK1_MYSQL_SERVER="yes"

    NETWORK1_MYSQL_IN_CLI_IPADDR[0]=$ANY_IPADDR
    NETWORK1_MYSQL_IN_SRV_IPADDR[0]="192.168.1.1"
```

In the above example, we enable MySQL incoming client request on our internal subnet (**NETWORK1_MYSQL_SERVER="yes"**).

Next, we instruct the firewall that in our internal subnet (**NETWORK1**), all MySQL (**MYSQL**) incoming packets (**IN**) with source IP address (**SRC_IPADDR**) of any IP address (**$ANY_IPADDR**) and with destination IP address (**DST_IPADDR**) "192.168.1.1" will be allowed to come in and/or start a new connection.

In other terms, any incoming MySQL packets with any IP address as source IP address and with 192.168.1.1 as destination IP address will be allowed to come in and/or start a new connection.

Using the connection tracking capability of IPTables, the related MySQL outgoing packets are automatically allowed back out by the firewall. In this case, our machine from our internal subnet that has the IP address 192.168.1.1 is the MySQL server and it is allowed to receive requests from any MySQL client from the Internet.

NOTE: The MySQL client from the Internet thinks that it talks to our Gateway server, so the actual destination IP address of the packet is our external interface IP address (**$INTERFACE1_IPADDR**), but the packet is automatically DNATed to 192.168.1.1.

Pay special attention to the above parameters. We noted that IP address "192.168.1.1" is used as the value for the incoming client requests with the forwarding feature. This is important, if your internal workstation IP address is different, you will have to adjust the setting to fit your own IP address for each of the forwarding definitions.

Step3
Now that our parameters for MySQL service have been correctly entered in the GIPTables configuration file, we need to restart our GIPTables firewall for the changes to take effect.

- To restart GIPTables on your system, use the following command:
 `[root@deep /]# /etc/init.d/giptables restart`

Well, now we have a better idea about what these cryptic definitions do and how to change them to fit our needs depending of the type of firewall that we need for our server. Human error is inevitable and if we entered all the additional parameters into GIPTables by hand, we could in inadvertently make some errors. To avoid this risk, GIPTables provides through it's "giptables.conf.README" file all the possible definitions for available services that can be used with it.

Therefore, if you need to add some additional services, which do not exist by default in the giptables.conf file, you can refer to this file to get the parameters to make your service run with GIPTables Firewall. All you'll need to do is to cut and paste the required lines into your GIPTables configuration file and set up each parameter by answering "yes" or "no" to the questions.

Running the type of GIPTables firewall that you need

All servers should be configured to block the unused ports, **even if they are not a firewall server**. This is required for increased security. Imagine that someone gains access to your main firewall server: if your other servers are not configured to block unused ports, this can result a serious network security risk. The same is true for local connections; unauthorized employees can gain access to your other servers from inside the network.

As you should know now, before running GIPTables in your system, you must create a symbolic link under the /etc directory that points to the GIPTables configuration file suitable for your system. Once this configuration file exists under your /etc directory, all you have to do is to edit it and set-up your networking configuration to make it work for you. This is true with all of server types except for a Gateway Server which differs as explained below.

1) You may need to forward external traffic to your internal network.
2) You may need some specific services not available by default.
3) You need to use the `SNAT` feature of Linux.
4) You need to use the `DNAT` feature of Linux.

The `GIPTables` configuration file for a Gateway Server allows you to accomplish these special requirements but requires more work from your part. This is the reason why we will show you later both a complete example configuration file and the required steps for a Gateway/Proxy Server `GIPTables` configuration that should work for most users. It is important to note that the below example is only a base starting point since every ones needs are different, and the number of services running on specific servers may change from one person to another.

All the following steps and explanations are valid for a Gateway/Proxy Server. For any other type of server, you only need to create the symbolic link under your `/etc` directory that points to your type of server configuration and then start your firewall after setting up your networking configuration in the `giptables.conf` file.

Unlike other types of `GIPTables` firewall configuration file, e.g. a Web, Mail, DNS Servers, etc., configuring a Linux Server to masquerade and forward traffic from the inside private network that has unregistered IP addresses (i.e. `192.168.1.0/24`) to the outside network (i.e. the Internet) requires a special setup of your kernel and your `GIPTables` firewall configuration file. This kind of configuration is also known as a Gateway Server or Proxy Server (a machine that serves as a gateway for internal traffic to external traffic). This configuration must be set only if you have the need for this kind of service.

Some Points to Consider

You can assume that you are at risk if you connect your system to the Internet. Your gateway to the Internet is your greatest exposure, so we recommend the following:

- ✓ The Gateway should not run more applications than are absolutely necessary.
- ✓ The Gateway should strictly limit the type and number of protocols allowed to flow through it (protocols potentially provide security holes, such as FTP and telnet).
- ✓ Any system containing confidential or sensitive information should not be directly accessible from the Internet.
- ✓ A Proxy program like `Squid` is highly recommended on the Gateway Server.

The `GIPTables` configuration file for a Gateway/Proxy Server

Masquerading means that if one of the computers on your local network for which your Linux machine (or Gateway/Proxy) acts as a firewall wants to send something to the outside, your machine can "masquerade" as that computer. In other words, it forwards the traffic to the intended outside destination, but makes it look like it came from the firewall machine itself.

It works both ways: if the outside host replies, the Linux firewall will silently forward the traffic to the corresponding local computer. This way, the computers on your local network are completely invisible to the outside world, even though they can reach outside and can receive replies. This makes it possible to have the computers on the local network participate on the Internet even if they don’t have officially registered IP addresses.

Step1
The IP masquerading code will only work if IP forwarding is enabled on your system. This feature is by default disabled and you can enable it with the following command:

- To enable IPv4 forwarding on your Linux system, use the following command:
 Edit the **sysctl.conf** file (vi /etc/sysctl.conf) and add the following lines:

```
# Enable packet forwarding (required only for Gateway, VPN, Proxy, PPP)
net.ipv4.ip_forward = 1
```

You must restart your network for the change to take effect. The command to restart the network is:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

Step 2
Create the symbolic link to the **giptables.conf** file that points to the right GIPTables configuration file suitable for our setup of a Gateway Server.

- This can be accomplished with the following command:

```
[root@deep /]# ln -s /lib/giptables/conf/giptables.conf.gateway
/etc/giptables.conf
```

Step3
Once the symbolic link is created, we will edit it to suit our requirements. The text in bold are the parts of the configuration that must be modified to satisfy your needs.

This is the configuration script file for a Gateway/Proxy Server, it will:

1 Log and limit the amount of incoming dropped packets.
2 Implement the Syn-flood protection.
3 Make sure that all NEW incoming tcp connections are SYN packets.
4 Protect from Spoofing and bad addresses.
5 Allow DNS client requests on internal & external interfaces.
6 Allow FTP client requests on internal & external interfaces.
7 Allow SSH client requests on internal & external interfaces.
8 Allow SSH server requests on the external interface.
9 Allow SMTP client requests on internal & external interfaces.
10 Allow POP3 client requests on the internal interface.
11 Allow POP3S client requests on the internal interface.
12 Allow HTTP client requests on internal & external interfaces.
13 Allow HTTPS client requests on internal & external interfaces.
14 Allow SQUID client requests on internal & external interfaces.
15 Allow SQUID server requests on the external interface.
16 Allow NETBIOS client requests on the internal interface.
17 Allow NETBIOS server requests on the internal interface.
18 Allow TRACEROUTE client requests on internal & external interfaces.
19 Allow TRACEROUTE server requests on internal & external interfaces.
20 Allow ICMP client requests on internal & external interfaces.
21 Allow DHCP client requests on the internal interface.

If you don't want some of the services listed in the firewall rules files for the Gateway/Proxy Server, disable them by saying "no" to the questions. If you want some other services that are not enabled, simply say, "yes" to the questions. If the service does not exist, add it to your configuration based on the available examples from the giptables.conf.README file.

- To edit the **giptales.conf** file, use the following command:
 [root@deep /]# **vi /etc/giptables.conf**

```
# ------------------------------------------------------------------------------
# GIPTables Firewall v1.1 http://www.giptables.org
# Copyright (C) 2002 Adrian Pascalau <apascalau@openna.com>
# GATEWAY main configuration file
#
# ------------------------------------------------------------------------------
# This file is part of GIPTables Firewall
#
# GIPTables Firewall is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA

# ------------------------------------------------------------------------------
# DEBUG
#

    DEBUG="off"

# ------------------------------------------------------------------------------
# Some definitions for easy maintenance
# Edit these to suit your system
#

    MONOLITIC_KERNEL="no"

# Interface 0: This is our external network interface
# It is directly connected to Internet

    INTERFACE0="eth0"
    INTERFACE0_IPADDR="x.x.x.x"
    ANY_IPADDR="0/0"

# Interface 1: This is our internal network interface
# It is directly connected to our internal Network 1

    INTERFACE1="eth1"
    INTERFACE1_IPADDR="192.168.1.254"
    NETWORK1="192.168.1.0/24"

# Do you need Network Address Translation for your internal network?

    NETWORK1_NAT="yes"
```

```
# Your name servers ip address

    ISP_PRIMARY_DNS_SERVER="y.y.y.y"
    ISP_SECONDARY_DNS_SERVER="z.z.z.z"

# SYSLOG server ip address

    SYSLOG_SERVER="c.c.c.c"

# Loopback interface

    LOOPBACK_INTERFACE="lo"                     # Loopback interface

# Port declarations, do not change them

    PRIV_PORTS="0:1023"
    UNPRIV_PORTS="1024:65535"

# --------------------------------------------------------------------------
# Loading custom firewall rules from /etc/rc.d/rc.giptables.custom
#

    LOAD_CUSTOM_RULES="yes"

# --------------------------------------------------------------------------
# Logging
# Limit the amount of incoming dropped packets that gets sent to the logs
#

# We log & drop all the packets that are not expected. In order to avoid
# our logs beeing flooded, we rate limit the logging

# Interface 0 log dropped packets

    INTERFACE0_LOG_DROPPED_PACKETS="yes"
    INTERFACE0_LOG_LIMIT="5/m"
    INTERFACE0_LOG_LIMIT_BURST="7"

# Interface 1 log dropped packets

    INTERFACE1_LOG_DROPPED_PACKETS="yes"
    INTERFACE1_LOG_LIMIT="7/m"
    INTERFACE1_LOG_LIMIT_BURST="9"

# Network 1 log forwarded dropped packets

    NETWORK1_LOG_DROPPED_PACKETS="yes"
    NETWORK1_LOG_LIMIT="9/m"
    NETWORK1_LOG_LIMIT_BURST="11"

# --------------------------------------------------------------------------
# Network Ghouls
# Refuse any connection from problem sites
#

# The /etc/rc.d/rc.giptables.blocked file contains a list of ip addresses that
# will be blocked from having any kind of access to your server on all your
# interfaces if the next option is "yes"

    NETWORK_GHOULS="yes"
```

```
# ------------------------------------------------------------------------------
# Syn-flood protection
# Limit the number of incoming tcp connections
#

    SYN_FLOOD_PROTECTION="yes"

# Interface 0 incoming syn-flood protection

    INTERFACE0_IN_SYN_FLOOD_PROTECTION="yes"
    INTERFACE0_IN_TCP_CONN_LIMIT="1/s"
    INTERFACE0_IN_TCP_CONN_LIMIT_BURST="3"

# Interface 1 incoming syn-flood protection

    INTERFACE1_IN_SYN_FLOOD_PROTECTION="yes"
    INTERFACE1_IN_TCP_CONN_LIMIT="3/s"
    INTERFACE1_IN_TCP_CONN_LIMIT_BURST="5"

# Network 1 forwarded incoming syn-flood protection

    NETWORK1_IN_SYN_FLOOD_PROTECTION="yes"
    NETWORK1_IN_TCP_CONN_LIMIT="5/s"
    NETWORK1_IN_TCP_CONN_LIMIT_BURST="7"

# ------------------------------------------------------------------------------
# Sanity check
#

    SANITY_CHECK="yes"

# Make sure NEW incoming tcp connections are SYN packets

    INTERFACE0_IN_DROP_NEW_WITHOUT_SYN="yes"
    INTERFACE1_IN_DROP_NEW_WITHOUT_SYN="yes"
    NETWORK1_IN_DROP_NEW_WITHOUT_SYN="yes"

# Drop all incoming fragments

    INTERFACE0_IN_DROP_ALL_FRAGMENTS="yes"
    INTERFACE1_IN_DROP_ALL_FRAGMENTS="yes"
    NETWORK1_IN_DROP_ALL_FRAGMENTS="yes"

# Drop all incoming malformed XMAS packets

    INTERFACE0_IN_DROP_XMAS_PACKETS="yes"
    INTERFACE1_IN_DROP_XMAS_PACKETS="yes"
    NETWORK1_IN_DROP_XMAS_PACKETS="yes"

# Drop all incoming malformed NULL packets

    INTERFACE0_IN_DROP_NULL_PACKETS="yes"
    INTERFACE1_IN_DROP_NULL_PACKETS="yes"
    NETWORK1_IN_DROP_NULL_PACKETS="yes"
```

```
# ------------------------------------------------------------------------
# Spoofing and bad addresses
#

    REFUSE_SPOOFING="yes"

# Refuse incoming packets claiming to be from the ip addresses of our
interfaces

    REFUSE_SPOOFING_IPADDR[0]=$INTERFACE0_IPADDR
    INTERFACE0_IN_REFUSE_SPOOFING[0]="yes"
    INTERFACE1_IN_REFUSE_SPOOFING[0]="no"
    NETWORK1_IN_REFUSE_SPOOFING[0]="yes"

    REFUSE_SPOOFING_IPADDR[1]=$INTERFACE1_IPADDR
    INTERFACE0_IN_REFUSE_SPOOFING[1]="no"
    INTERFACE1_IN_REFUSE_SPOOFING[1]="yes"
    NETWORK1_IN_REFUSE_SPOOFING[1]="no"

# Refuse incoming packets claiming to be from broadcast-src address range

    REFUSE_SPOOFING_IPADDR[2]="0.0.0.0/8"

# If you provide DHCP services on one of your interfaces, do not forget to
# set the following option related to that interface to "no"

    INTERFACE0_IN_REFUSE_SPOOFING[2]="yes"
    INTERFACE1_IN_REFUSE_SPOOFING[2]="no"
    NETWORK1_IN_REFUSE_SPOOFING[2]="yes"

# Refuse incoming packets claiming to be from reserved loopback address range

    REFUSE_SPOOFING_IPADDR[3]="127.0.0.0/8"
    INTERFACE0_IN_REFUSE_SPOOFING[3]="yes"
    INTERFACE1_IN_REFUSE_SPOOFING[3]="yes"
    NETWORK1_IN_REFUSE_SPOOFING[3]="yes"

# Refuse incoming packets claiming to be from class A private network

    REFUSE_SPOOFING_IPADDR[4]="10.0.0.0/8"
    INTERFACE0_IN_REFUSE_SPOOFING[4]="yes"
    INTERFACE1_IN_REFUSE_SPOOFING[4]="yes"
    NETWORK1_IN_REFUSE_SPOOFING[4]="yes"

# Refuse incoming packets claiming to be from class B private network

    REFUSE_SPOOFING_IPADDR[5]="172.16.0.0/12"
    INTERFACE0_IN_REFUSE_SPOOFING[5]="yes"
    INTERFACE1_IN_REFUSE_SPOOFING[5]="yes"
    NETWORK1_IN_REFUSE_SPOOFING[5]="yes"

# Refuse incoming packets claiming to be from class C private network

    REFUSE_SPOOFING_IPADDR[6]="192.168.0.0/16"
    INTERFACE0_IN_REFUSE_SPOOFING[6]="yes"
    INTERFACE1_IN_REFUSE_SPOOFING[6]="no"
    NETWORK1_IN_REFUSE_SPOOFING[6]="yes"
```

```
# Refuse incoming packets claiming to be from class D, E, and unallocated

    REFUSE_SPOOFING_IPADDR[7]="224.0.0.0/3"
    INTERFACE0_IN_REFUSE_SPOOFING[7]="yes"
    INTERFACE1_IN_REFUSE_SPOOFING[7]="yes"
    NETWORK1_IN_REFUSE_SPOOFING[7]="yes"

# ******************************************************************************
#                                                                              *
#                                    A N Y                                     *
#                                                                              *
# ******************************************************************************

    ACCEPT_ANY="no"

# ******************************************************************************
#                                                                              *
#                                    D N S                                     *
#                                                                              *
# ******************************************************************************

    ACCEPT_DNS="yes"

# ------------------------------------------------------------------------------
# DNS outgoing client request
#

# Interface 0 DNS outgoing client request

    INTERFACE0_DNS_CLIENT="yes"

    INTERFACE0_DNS_OUT_SRC_IPADDR[0]=$INTERFACE0_IPADDR
    INTERFACE0_DNS_OUT_DST_IPADDR[0]=$ISP_PRIMARY_DNS_SERVER
    INTERFACE0_DNS_OUT_UDP_REQUEST[0]="yes"
    INTERFACE0_DNS_OUT_TCP_REQUEST[0]="yes"
    INTERFACE0_DNS_OUT_SPORT53_REQUEST[0]="no"

    INTERFACE0_DNS_OUT_SRC_IPADDR[1]=$INTERFACE0_IPADDR
    INTERFACE0_DNS_OUT_DST_IPADDR[1]=$ISP_SECONDARY_DNS_SERVER
    INTERFACE0_DNS_OUT_UDP_REQUEST[1]="yes"
    INTERFACE0_DNS_OUT_TCP_REQUEST[1]="yes"
    INTERFACE0_DNS_OUT_SPORT53_REQUEST[1]="no"

# Network 1 DNS forwarded outgoing client request

    NETWORK1_DNS_CLIENT="yes"

    NETWORK1_DNS_OUT_SRC_IPADDR[0]=$NETWORK1
    NETWORK1_DNS_OUT_DST_IPADDR[0]=$ISP_PRIMARY_DNS_SERVER
    NETWORK1_DNS_OUT_UDP_REQUEST[0]="yes"
    NETWORK1_DNS_OUT_TCP_REQUEST[0]="yes"
    NETWORK1_DNS_OUT_SPORT53_REQUEST[0]="no"

# ******************************************************************************
#                                                                              *
#                                    F T P                                     *
#                                                                              *
# ******************************************************************************

    ACCEPT_FTP="yes"
```

```
# ------------------------------------------------------------------------
# FTP outgoing client request
#

# Interface 0 FTP outgoing client request

    INTERFACE0_FTP_CLIENT="yes"

    INTERFACE0_FTP_OUT_SRC_IPADDR[0]=$INTERFACE0_IPADDR
    INTERFACE0_FTP_OUT_DST_IPADDR[0]=$ANY_IPADDR
    INTERFACE0_FTP_OUT_PASIVE[0]="yes"
    INTERFACE0_FTP_OUT_ACTIVE[0]="no"

# Network 1 FTP forwarded outgoing client request

    NETWORK1_FTP_CLIENT="yes"

    NETWORK1_FTP_OUT_SRC_IPADDR[0]=$NETWORK1
    NETWORK1_FTP_OUT_DST_IPADDR[0]=$ANY_IPADDR
    NETWORK1_FTP_OUT_PASIVE[0]="yes"
    NETWORK1_FTP_OUT_ACTIVE[0]="yes"

# ************************************************************************
#                                                                        *
#                                  S S H                                 *
#                                                                        *
# ************************************************************************

    ACCEPT_SSH="yes"

# ------------------------------------------------------------------------
# SSH outgoing client request
#

# Interface 0 SSH outgoing client request

    INTERFACE0_SSH_CLIENT="yes"

    INTERFACE0_SSH_OUT_SRC_IPADDR[0]=$INTERFACE0_IPADDR
    INTERFACE0_SSH_OUT_DST_IPADDR[0]=$ANY_IPADDR

# Network 1 SSH forwarded outgoing client request

    NETWORK1_SSH_CLIENT="yes"

    NETWORK1_SSH_OUT_SRC_IPADDR[0]=$NETWORK1
    NETWORK1_SSH_OUT_DST_IPADDR[0]=$ANY_IPADDR

# ------------------------------------------------------------------------
# SSH incoming client request
#

# Interface 0 SSH incoming client request

    INTERFACE0_SSH_SERVER="yes"

    INTERFACE0_SSH_IN_SRC_IPADDR[0]=$ANY_IPADDR
    INTERFACE0_SSH_IN_DST_IPADDR[0]=$INTERFACE0_IPADDR
```

```
# Interface 1 SSH incoming client request

    INTERFACE1_SSH_SERVER="yes"

    INTERFACE1_SSH_IN_SRC_IPADDR[0]=$NETWORK1
    INTERFACE1_SSH_IN_DST_IPADDR[0]=$INTERFACE0_IPADDR

# ************************************************************************
#                                                                        *
#                              T E L N E T                               *
#                                                                        *
# ************************************************************************

    ACCEPT_TELNET="no"

# ------------------------------------------------------------------------
# TELNET outgoing client request
#

# Interface 0 TELNET outgoing client request

    INTERFACE0_TELNET_CLIENT="yes"

    INTERFACE0_TELNET_OUT_SRC_IPADDR[0]=$INTERFACE0_IPADDR
    INTERFACE0_TELNET_OUT_DST_IPADDR[0]=$ANY_IPADDR

# Network 1 TELNET forwarded outgoing client request

    NETWORK1_TELNET_CLIENT="yes"

    NETWORK1_TELNET_OUT_SRC_IPADDR[0]=$NETWORK1
    NETWORK1_TELNET_OUT_DST_IPADDR[0]=$ANY_IPADDR

# ------------------------------------------------------------------------
# TELNET incoming client request
#

# Interface 1 TELNET incoming client request

    INTERFACE1_TELNET_SERVER="no"

    INTERFACE1_TELNET_IN_SRC_IPADDR[0]=$NETWORK1
    INTERFACE1_TELNET_IN_DST_IPADDR[0]=$INTERFACE0_IPADDR

# ************************************************************************
#                                                                        *
#                             T E L N E T S                              *
#                                                                        *
# ************************************************************************

    ACCEPT_TELNETS="no"

# ************************************************************************
#                                                                        *
#                               S M T P                                  *
#                                                                        *
# ************************************************************************

    ACCEPT_SMTP="yes"
```

```
# ------------------------------------------------------------------------
# SMTP outgoing client request
#

# Interface 0 SMTP outgoing client request

    INTERFACE0_SMTP_CLIENT="yes"

    INTERFACE0_SMTP_OUT_SRC_IPADDR[0]=$INTERFACE0_IPADDR
    INTERFACE0_SMTP_OUT_DST_IPADDR[0]=$ANY_IPADDR

# Network 1 SMTP forwarded outgoing client request

    NETWORK1_SMTP_CLIENT="yes"

    NETWORK1_SMTP_OUT_SRC_IPADDR[0]=$NETWORK1
    NETWORK1_SMTP_OUT_DST_IPADDR[0]=$ANY_IPADDR

# ************************************************************************
#                                                                        *
#                                S M T P S                               *
#                                                                        *
# ************************************************************************

    ACCEPT_SMTPS="no"

# ------------------------------------------------------------------------
# SMTPS outgoing client request
#

# Interface 0 SMTPS outgoing client request

    INTERFACE0_SMTPS_CLIENT="yes"

    INTERFACE0_SMTPS_OUT_SRC_IPADDR[0]=$INTERFACE0_IPADDR
    INTERFACE0_SMTPS_OUT_DST_IPADDR[0]=$ANY_IPADDR

# Network 1 SMTPS forwarded outgoing client request

    NETWORK1_SMTPS_CLIENT="yes"

    NETWORK1_SMTPS_OUT_SRC_IPADDR[0]=$NETWORK1
    NETWORK1_SMTPS_OUT_DST_IPADDR[0]=$ANY_IPADDR

# ************************************************************************
#                                                                        *
#                                P O P 3                                 *
#                                                                        *
# ************************************************************************


    ACCEPT_POP3="yes"

# ------------------------------------------------------------------------
# POP3 outgoing client request
#

# Network 1 POP3 forwarded outgoing client request

    NETWORK1_POP3_CLIENT="yes"

    NETWORK1_POP3_OUT_SRC_IPADDR[0]=$NETWORK1
    NETWORK1_POP3_OUT_DST_IPADDR[0]=$ANY_IPADDR
```

```
# ************************************************************************
#                                                                        *
#                                P O P 3 S                               *
#                                                                        *
# ************************************************************************

    ACCEPT_POP3S="yes"

# ------------------------------------------------------------------------
# POP3S outging client request
#

# Network 1 POP3S forwarded outging client request

    NETWORK1_POP3S_CLIENT="yes"

    NETWORK1_POP3S_OUT_SRC_IPADDR[0]=$NETWORK1
    NETWORK1_POP3S_OUT_DST_IPADDR[0]=$ANY_IPADDR

# ************************************************************************
#                                                                        *
#                                I M A P                                 *
#                                                                        *
# ************************************************************************

    ACCEPT_IMAP="no"

# ------------------------------------------------------------------------
# IMAP outgoing client request
#

# Network 1 IMAP forwarded outgoing client request

    NETWORK1_IMAP_CLIENT="yes"

    NETWORK1_IMAP_OUT_SRC_IPADDR[0]=$NETWORK1
    NETWORK1_IMAP_OUT_DST_IPADDR[0]=$ANY_IPADDR

# ************************************************************************
#                                                                        *
#                                I M A P S                               *
#                                                                        *
# ************************************************************************

    ACCEPT_IMAPS="no"

# ------------------------------------------------------------------------
# IMAPS outgoing client request
#

# Network 1 IMAPS forwarded outgoing client request

    NETWORK1_IMAPS_CLIENT="yes"

    NETWORK1_IMAPS_OUT_SRC_IPADDR[0]=$NETWORK1
    NETWORK1_IMAPS_OUT_DST_IPADDR[0]=$ANY_IPADDR

# ************************************************************************
#                                                                        *
#                                H T T P                                 *
#                                                                        *
# ************************************************************************
```

```
    ACCEPT_HTTP="yes"

# --------------------------------------------------------------------------
# HTTP outgoing client request
#

# Interface 0 HTTP outgoing client request

    INTERFACE0_HTTP_CLIENT="yes"

    INTERFACE0_HTTP_OUT_SRC_IPADDR[0]=$INTERFACE0_IPADDR
    INTERFACE0_HTTP_OUT_DST_IPADDR[0]=$ANY_IPADDR

# Network 1 HTTP forwarded outgoing client request

    NETWORK1_HTTP_CLIENT="yes"

    NETWORK1_HTTP_OUT_SRC_IPADDR[0]=$NETWORK1
    NETWORK1_HTTP_OUT_DST_IPADDR[0]=$ANY_IPADDR

# ************************************************************************
#                                                                        *
#                                H T T P S                               *
#                                                                        *
# ************************************************************************

    ACCEPT_HTTPS="yes"

# --------------------------------------------------------------------------
# HTTPS outgoing client request
#

# Interface 0 HTTPS outgoing client request

    INTERFACE0_HTTPS_CLIENT="yes"

    INTERFACE0_HTTPS_OUT_SRC_IPADDR[0]=$INTERFACE0_IPADDR
    INTERFACE0_HTTPS_OUT_DST_IPADDR[0]=$ANY_IPADDR

# Network 1 HTTPS forwarded outgoing client request

    NETWORK1_HTTPS_CLIENT="yes"

    NETWORK1_HTTPS_OUT_SRC_IPADDR[0]=$NETWORK1
    NETWORK1_HTTPS_OUT_DST_IPADDR[0]=$ANY_IPADDR

# ************************************************************************
#                                                                        *
#                                S Q U I D                               *
#                                                                        *
# ************************************************************************

    ACCEPT_SQUID="yes" # Squid in Proxy-Caching Mode

# --------------------------------------------------------------------------
# SQUID outgoing client request
#

# Interface 0 SQUID outgoing client request

    INTERFACE0_SQUID_CLIENT="yes"
```

```
    INTERFACE0_SQUID_OUT_SRC_IPADDR[0]=$INTERFACE0_IPADDR
    INTERFACE0_SQUID_OUT_DST_IPADDR[0]=$ANY_IPADDR

# Interface 1 SQUID outgoing client request

    INTERFACE1_SQUID_CLIENT="yes"

    INTERFACE1_SQUID_OUT_SRC_IPADDR[0]=$INTERFACE1_IPADDR
    INTERFACE1_SQUID_OUT_DST_IPADDR[0]=$NETWORK1

# Network 1 SQUID forwarded outgoing client request

    NETWORK1_SQUID_CLIENT="yes"

    NETWORK1_SQUID_OUT_SRC_IPADDR[0]=$NETWORK1
    NETWORK1_SQUID_OUT_DST_IPADDR[0]=$ANY_IPADDR

# ------------------------------------------------------------------------
# SQUID incoming client request
#

# Interface 0 SQUID incoming client request

    INTERFACE0_SQUID_SERVER="yes"

    INTERFACE0_SQUID_IN_SRC_IPADDR[0]=$ANY_IPADDR
    INTERFACE0_SQUID_IN_DST_IPADDR[0]=$INTERFACE0_IPADDR

# Interface 1 SQUID incoming client request

    INTERFACE1_SQUID_SERVER="yes"

    INTERFACE1_SQUID_IN_SRC_IPADDR[0]=$NETWORK1
    INTERFACE1_SQUID_IN_DST_IPADDR[0]=$INTERFACE1_IPADDR

# ************************************************************************
#                                                                        *
#                            W E B C A C H E                             *
#                                                                        *
# ************************************************************************

    ACCEPT_WEBCACHE="no" # Squid in HTTPD-Accelerator Mode

# ************************************************************************
#                                                                        *
#                                N N T P                                 *
#                                                                        *
# ************************************************************************

    ACCEPT_NNTP="no"

# ------------------------------------------------------------------------
# NNTP outgoing client request
#

# Network 1 NNTP forwarded outgoing client request

    NETWORK1_NNTP_CLIENT="yes"

    NETWORK1_NNTP_OUT_SRC_IPADDR[0]=$NETWORK1
    NETWORK1_NNTP_OUT_DST_IPADDR[0]=$ANY_IPADDR
```

```
# ************************************************************************
#                                                                        *
#                                N N T P S                               *
#                                                                        *
# ************************************************************************

    ACCEPT_NNTPS="no"

# ************************************************************************
#                                                                        *
#                                M Y S Q L                               *
#                                                                        *
# ************************************************************************

    ACCEPT_MYSQL="no"

# ************************************************************************
#                                                                        *
#                            P O S T G R E S                             *
#                                                                        *
# ************************************************************************

    ACCEPT_POSTGRES="no"

# ************************************************************************
#                                                                        *
#                              O R A C L E                               *
#                                                                        *
# ************************************************************************

    ACCEPT_ORACLE="no"

# ************************************************************************
#                                                                        *
#                                L D A P                                 *
#                                                                        *
# ************************************************************************

    ACCEPT_LDAP="no"

# ************************************************************************
#                                                                        *
#                                L D A P S                               *
#                                                                        *
# ************************************************************************

    ACCEPT_LDAPS="no"

# ************************************************************************
#                                                                        *
#                                A U T H                                 *
#                                                                        *
# ************************************************************************

    ACCEPT_AUTH="no"

# ------------------------------------------------------------------------
# AUTH outgoing client request
#

# Interface 0 AUTH outgoing client request
```

```
    INTERFACE0_AUTH_CLIENT="yes"

    INTERFACE0_AUTH_OUT_SRC_IPADDR[0]=$INTERFACE0_IPADDR
    INTERFACE0_AUTH_OUT_DST_IPADDR[0]=$ANY_IPADDR

# Network 1 AUTH forwarded outgoing client request

    NETWORK1_AUTH_CLIENT="yes"

    NETWORK1_AUTH_OUT_SRC_IPADDR[0]=$NETWORK1
    NETWORK1_AUTH_OUT_DST_IPADDR[0]=$ANY_IPADDR

# ***************************************************************************
#                                                                           *
#                                W H O I S                                  *
#                                                                           *
# ***************************************************************************

    ACCEPT_WHOIS="no"

# ---------------------------------------------------------------------------
# WHOIS outgoing client request
#

# Interface 0 WHOIS outgoing client request

    INTERFACE0_WHOIS_CLIENT="yes"

    INTERFACE0_WHOIS_OUT_SRC_IPADDR[0]=$INTERFACE0_IPADDR
    INTERFACE0_WHOIS_OUT_DST_IPADDR[0]=$ANY_IPADDR

# Network 1 WHOIS forwarded outgoing client request

    NETWORK1_WHOIS_CLIENT="yes"

    NETWORK1_WHOIS_OUT_SRC_IPADDR[0]=$NETWORK1
    NETWORK1_WHOIS_OUT_DST_IPADDR[0]=$ANY_IPADDR

# ***************************************************************************
#                                                                           *
#                              F I N G E R                                  *
#                                                                           *
# ***************************************************************************

    ACCEPT_FINGER="no"

# ---------------------------------------------------------------------------
# FINGER outgoing client request
#

# Interface 0 FINGER outgoing client request

    INTERFACE0_FINGER_CLIENT="yes"

    INTERFACE0_FINGER_OUT_SRC_IPADDR[0]=$INTERFACE0_IPADDR
    INTERFACE0_FINGER_OUT_DST_IPADDR[0]=$ANY_IPADDR

# Network 1 FINGER forwarded outgoing client request

    NETWORK1_FINGER_CLIENT="yes"

    NETWORK1_FINGER_OUT_SRC_IPADDR[0]=$NETWORK1
    NETWORK1_FINGER_OUT_DST_IPADDR[0]=$ANY_IPADDR
```

```
# ********************************************************************************
#                                                                                *
#                                    N T P                                       *
#                                                                                *
# ********************************************************************************

    ACCEPT_NTP="no"

# --------------------------------------------------------------------------------
# NTP outgoing client request
#

# Interface 0 NTP outgoing client request

    INTERFACE0_NTP_CLIENT="yes"

    INTERFACE0_NTP_OUT_SRC_IPADDR[0]=$INTERFACE0_IPADDR
    INTERFACE0_NTP_OUT_DST_IPADDR[0]=$ANY_IPADDR

# Network 1 NTP forwarded outgoing client request

    NETWORK1_NTP_CLIENT="no"

    NETWORK1_NTP_OUT_SRC_IPADDR[0]=$NETWORK1
    NETWORK1_NTP_OUT_DST_IPADDR[0]=$ANY_IPADDR

# ********************************************************************************
#                                                                                *
#                                N E T B I O S                                   *
#                                                                                *
# ********************************************************************************

    ACCEPT_NETBIOS="yes"

# --------------------------------------------------------------------------------
# NETBIOS outgoing client request
#

# Interface 1 NETBIOS outgoing client request

    INTERFACE1_NETBIOS_CLIENT="yes"

    INTERFACE1_NETBIOS_OUT_SRC_IPADDR[0]=$INTERFACE1_IPADDR
    INTERFACE1_NETBIOS_OUT_DST_IPADDR[0]=$NETWORK1

# --------------------------------------------------------------------------------
# NETBIOS incoming client request
#

# Interface 1 NETBIOS incoming client request

    INTERFACE1_NETBIOS_SERVER="yes"

    INTERFACE1_NETBIOS_IN_SRC_IPADDR[0]=$NETWORK1
    INTERFACE1_NETBIOS_IN_DST_IPADDR[0]=$INTERFACE1_IPADDR

# ********************************************************************************
#                                                                                *
#                                S Y S L O G                                     *
#                                                                                *
# ********************************************************************************
```

```
    ACCEPT_SYSLOG="no"

# ------------------------------------------------------------------------
# SYSLOG outgoing client request
#

# Interface 1 SYSLOG outgoing client request

    INTERFACE1_SYSLOG_CLIENT="yes"

    INTERFACE1_SYSLOG_OUT_SRC_IPADDR[0]=$INTERFACE1_IPADDR
    INTERFACE1_SYSLOG_OUT_DST_IPADDR[0]=$SYSLOG_SERVER

# ************************************************************************
#                                                                        *
#                           T R A C E R O U T E                          *
#                                                                        *
# ************************************************************************

    ACCEPT_TRACEROUTE="yes"

# ------------------------------------------------------------------------
# TRACEROUTE outgoing client request
#

# Interface 0 TRACEROUTE outgoing client request

    INTERFACE0_TRACEROUTE_CLIENT="yes"

    INTERFACE0_TRACEROUTE_OUT_SRC_IPADDR[0]=$INTERFACE0_IPADDR
    INTERFACE0_TRACEROUTE_OUT_DST_IPADDR[0]=$ANY_IPADDR

# Network 1 TRACEROUTE forwarded outgoing client request

    NETWORK1_TRACEROUTE_CLIENT="yes"

    NETWORK1_TRACEROUTE_OUT_SRC_IPADDR[0]=$NETWORK1
    NETWORK1_TRACEROUTE_OUT_DST_IPADDR[0]=$ANY_IPADDR

# ------------------------------------------------------------------------
# TRACEROUTE incoming client request
#

# Interface 1 TRACEROUTE incoming client request

    INTERFACE1_TRACEROUTE_SERVER="no"

    INTERFACE1_TRACEROUTE_IN_SRC_IPADDR[0]=$NETWORK1
    INTERFACE1_TRACEROUTE_IN_DST_IPADDR[0]=$INTERFACE0_IPADDR

# ************************************************************************
#                                                                        *
#                                 I C M P                                *
#                                                                        *
# ************************************************************************

    ACCEPT_ICMP="yes"

# ------------------------------------------------------------------------
# ICMP outgoing client request
#

# Interface 0 ICMP outgoing client request
```

```
    INTERFACE0_ICMP_CLIENT="yes"

    INTERFACE0_ICMP_OUT_SRC_IPADDR[0]=$INTERFACE0_IPADDR
    INTERFACE0_ICMP_OUT_DST_IPADDR[0]=$ANY_IPADDR

# Network 1 ICMP forwarded outgoing client request

    NETWORK1_ICMP_CLIENT="yes"

    NETWORK1_ICMP_OUT_SRC_IPADDR[0]=$NETWORK1
    NETWORK1_ICMP_OUT_DST_IPADDR[0]=$ANY_IPADDR

# ------------------------------------------------------------------------
# ICMP incoming client request
#

# Interface 1 ICMP incoming client request

    INTERFACE1_ICMP_SERVER="no"

    INTERFACE1_ICMP_IN_SRC_IPADDR[0]=$NETWORK1
    INTERFACE1_ICMP_IN_DST_IPADDR[0]=$INTERFACE0_IPADDR

    INTERFACE1_ICMP_IN_SRC_IPADDR[1]=$NETWORK1
    INTERFACE1_ICMP_IN_DST_IPADDR[1]=$INTERFACE1_IPADDR

# ************************************************************************
#                                                                        *
#                                D H C P                                 *
#                                                                        *
# ************************************************************************

    ACCEPT_DHCP="yes"

# ------------------------------------------------------------------------
# DHCP incoming client request
#

# Interface 1 DHCP incoming client request

    INTERFACE1_DHCP_SERVER="yes"

# If above option is "yes", do not forget to configure the following
# lines in the "Spoofing and bad addresses" section
# REFUSE_SPOOFING_IPADDR[2]="0.0.0.0/8"
# INTERFACE1_IN_REFUSE_SPOOFING[2]="no"

    INTERFACE1_DHCP_IN_SRC_IPADDR[0]=$NETWORK1
    INTERFACE1_DHCP_IN_DST_IPADDR[0]=$INTERFACE1_IPADDR

# ************************************************************************
#                                                                        *
#                                  E N D                                 *
#                                                                        *
# ************************************************************************

    DROP_EVERYTHING_FROM_HERE="yes"

# ------------------------------------------------------------------------
# LOG & DROP everything from here... just in case.
#
```

```
    INTERFACE0_IN_DROP_EVERYTHING_FROM_HERE="yes"
    INTERFACE1_IN_DROP_EVERYTHING_FROM_HERE="yes"
    NETWORK1_IN_DROP_EVERYTHING_FROM_HERE="yes"

# ------------------------------------------------------------------------
# End of file
```

Step 4
Once the configuration file has been configured, it is time to start the firewall on your system.

- To start the firewall on your system, use the following command:
```
[root@deep /]# /etc/init.d/giptables start
Starting Firewalling Services:           [OK]
```

`GIPTables-Firewall` Administrative Tools

The commands listed below are some that we use often, but many more exist. Check the manual page and documentation for more information.

`IPTables`

The `iptables` tool is used for the firewall packet filter administration of the system. We can use it to set up a firewall rules file, as we are doing in this book. Once firewall rules have been created we can play with its many commands to maintain, and inspect the rules in the kernel.

- To list all rules in the selected chain, use the command:
```
[root@deep /]# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
```

This command will list all rules in the selected chain. If no chain is selected, all chains are listed.

- To list all INPUT rules in the selected chain, use the command:
```
[root@deep /]# iptables -L INPUT
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
ACCEPT     all  --  192.168.1.0/24       anywhere
DROP       all  --  204.254.45.9         anywhere
DROP       all  --  187.231.11.5         anywhere
DROP       all  --  207.35.78.5          anywhere
```

- To list all OUTPUT rules in the selected chain, use the command:

```
[root@deep /]# iptables -L OUTPUT
Chain OUTPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             192.168.1.0/24
ACCEPT     udp  --  207.35.78.5          207.35.78.3        udp
spt:domain dpt:domain
ACCEPT     tcp  --  207.35.78.5          207.35.78.3        tcp
spts:1024:65535 dpt:domain
```

- To list all FORWARD rules in the selected chain, use the command:

```
[root@deep /]# iptables -L FORWARD
Chain FORWARD (policy DROP)
target     prot opt source               destination
DROP       tcp  --  anywhere             anywhere           tcp
DROP       tcp  --  anywhere             anywhere           tcp
DROP       all  -- !192.168.0.0/24       anywhere
ACCEPT     all  --  192.168.0.0/24       anywhere           state NEW
ACCEPT     all  -- !192.168.0.0/24       anywhere           state
```

This of course works only if you have configured Masquerading on your server (for Gateway servers in general).

- To list all rules in numeric OUTPUT in the selected chain, use the command:

```
[root@deep /]# iptables -nL
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     all  --  192.168.1.0/24       0.0.0.0/0
DROP       all  --  204.254.45.9         0.0.0.0/0

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     all  --  0.0.0.0/0            192.168.1.0/24
ACCEPT     udp  --  207.35.78.5          207.35.78.5
```

All the IP addresses and port numbers will be printed in numeric format.

CHAPTER 11

Squid Proxy Server

IN THIS CHAPTER

1. **Compiling - Optimizing & Installing `Squid`**
2. **Configuring `Squid`**
3. **Running `Squid` with Users Authentication Support**
4. **Securing `Squid`**
5. **Optimizing `Squid`**
6. **`Squid` Administrative Tools**
7. **The `cachemgr.cgi` program utility of `Squid`**

Linux Squid

Abstract

Another important program to consider is Squid especially for those of you that want to configure and run a Gateway Server for computers on your internal network and I know that you are numerous. Therefore, there is no doubt that for a Gateway Server set-up, IPTables, our Packet Filter software, and Squid, which will become our Application Gateway software, is required. In general, IPTables will protect our Gateway Server and Squid our private internal hosts. Do not install Squid on a Gateway Server without IPTables; both are very important and must be installed together if you want to have a secure Gateway system. IPTables is necessary to manage the legitimate opened ports on our server that Squid users will use to access the Internet or the network.

Proxy-servers like Squid, with their capability to save bandwidth, improve security, and increase web-surfing speeds are becoming more popular than ever. Currently only a few proxy-server programs are on the market. These proxy-servers have two main drawbacks: they are commercial, and they don't support ICP (ICP is used to exchange hints about the existence of URLs in neighbor caches). Squid is the best choice for a proxy-cache server since it is robust, free, and can use ICP features.

Derived from the "cached" software from the ARPA-funded Harvest research project, developed at the National Laboratory for Applied Network Research and funded by the National Science Foundation, Squid offers high-performance caching of web clients, and also supports FTP, Gopher, HTTP and HTTPS data objects.

It stores hot objects in RAM, maintains a robust database of objects on disk, has a complex access control mechanism (ACL), and supports the SSL protocol for proxying secure connections. In addition, it can be hierarchically linked to other Squid-based proxy servers for streamlined caching of pages through its unique ICP feature.

In our compilation and configuration we'll show you how to configure Squid depending on your needs. Two different set-ups are available.

The first will be to configure it to run as an **httpd-accelerator** to get more performance out of our Web Server. In accelerator mode, the Squid server acts as a reverse proxy cache: it accepts client requests, serves them out of cache, if possible, or requests them from the original Web Server for which it is the reverse proxy. However, this set-up is not what we need for a Gateway Server. It is only useful on a Web Server where you want better performance.

The second, the one suitable for our Gateway Server set-up will be to configure Squid as a **proxy-caching** server to be able to let all users on your corporate network use Squid to access the Internet. This is a very interesting addition when you run a Gateway Server your corporate network. A Gateway Server with IPTables as described earlier in this book plus a Squid Server mounted on it will highly improve the security and performance speed of the system. This is also the solution to control and restrict what can be viewed on the Internet.

With a Squid Server configured as a proxy-caching server on a Gateway Server, you will be able to block for example porno sites, underground sites, warez (if you want ☺), etc. many different possibilities exist, like authorizing access to the Internet based on specific hours or days.

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest Squid version number is 2.4.STABLE6

Packages

The following are based on information listed by Squid as of 2002/03/20. Please regularly check at www.squid-cache.org for the latest status. We chose to install the required component from source file because it provides the facility to fine tune the installation.

Source code is available from:
Squid Homepage: http://www.squid-cache.org/
Squid FTP Site: 206.168.0.9
You must be sure to download: squid-2.4.STABLE7-src.tar.gz

Though the procedures given in this chapter are likely to work on all Linux platforms, we have only tested it on OpenNA Linux and Red Hat Linux.

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all files installed into the system if the package is updated in the future. To solve this problem, it is a good idea to make a list of files on the system before you install Squid, and one afterwards, and then compares them using the **diff** utility of Linux to find out what files are placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep root]# find /* > Squid1
  ```
- And the following one after you install the software:
  ```
  [root@deep root]# find /* > Squid2
  ```
- Then use the following command to get a list of what changed:
  ```
  [root@deep root]# diff Squid1 Squid2 > Squid-Installed
  ```

By doing this, if in the future any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. We use the /root directory of the system to store all generated list files.

Compiling - Optimizing & Installing Squid

Below are the steps that you must make to configure, compile and optimize the Squid server software before installing it on your system. First off, we install the program as user 'root' so as to avoid any authorization problems.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- This can be done with the following commands:

```
[root@deep /]# cp squid-version-src.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf squid-version-src.tar.gz
```

Step 2
To avoid security risks, we must create a new user account called "squid" to be the owner of the Squid database cache files and daemon.

- To create this special Squid user on OpenNA Linux, use the following command:

```
[root@deep tmp]# groupadd -g 23 squid > /dev/null 2>&1 || :
[root@deep tmp]# useradd -c "Squid Server" -d /var/spool/squid -g 23 -s
/bin/false -u 23 squid > /dev/null 2>&1 || :
```

- To create this special Squid user on Red Hat Linux, use the following command:

```
[root@deep tmp]# useradd -c "Squid Server" -u 23 -s /bin/false -r -d
/var/spool/squid squid 2>/dev/null || :
```

The above command will create a null account, with no password, no valid shell, no files owned-nothing but a UID and a GID for the program. Remember that Squid daemon does not need to have a shell account on the server.

Step 3
After that, move into the newly created Squid source directory and perform the following steps to configure and optimize the software for your system.

- To move into the newly created Squid source directory use the command:

```
[root@deep tmp]# cd squid-2.4.STABLE7/
```

Step 4
There are some source files to modify before going into the configuration and compilation of the program; the changes allow us to fix some problems and to configure the program for our PATH environment variable under Linux.

- Edit the **acl.c** file (vi +651 src/acl.c) and change the line:

```
*Top = splay_insert(xstrdup(t), *Top, aclDomainCompare);
```

To read:

```
*Top = splay_insert(xstrdup(t), *Top, aclHostDomainCompare);
```

This fixes a small bug for our version of Linux.

- Edit the **Makefile.in** file (`vi +18 icons/Makefile.in`) and change the line:

```
DEFAULT_ICON_DIR  =  $(sysconfdir)/icons
```

To read:

```
DEFAULT_ICON_DIR  =  $(libexecdir)/icons
```

We change the variable (`sysconfdir`) to become (`libexecdir`). With this modification, the `/icons` directory of `Squid` will be located under the `/usr/lib/squid` directory.

- Edit the **Makefile.in** file (`vi +40 src/Makefile.in`) and change the lines:

```
DEFAULT_CACHE_LOG  =  $(localstatedir)/logs/cache.log
```

To read:

```
DEFAULT_CACHE_LOG  =  $(localstatedir)/log/squid/cache.log
```

```
DEFAULT_ACCESS_LOG  =  $(localstatedir)/logs/access.log
```

To read:

```
DEFAULT_ACCESS_LOG  =  $(localstatedir)/log/squid/access.log
```

```
DEFAULT_STORE_LOG  =  $(localstatedir)/logs/store.log
```

To read:

```
DEFAULT_STORE_LOG  =  $(localstatedir)/log/squid/store.log
```

```
DEFAULT_PID_FILE  =  $(localstatedir)/logs/squid.pid
```

To read:

```
DEFAULT_PID_FILE  =  $(localstatedir)/run/squid.pid
```

```
DEFAULT_SWAP_DIR  =  $(localstatedir)/cache
```

To read:

```
DEFAULT_SWAP_DIR  =  $(localstatedir)/spool/squid
```

```
DEFAULT_ICON_DIR  =  $(sysconfdir)/icons
```

To read:

```
DEFAULT_ICON_DIR  =  $(libexecdir)/icons
```

We change the default location of "cache.log", "access.log", and "store.log" files to be located under the /var/log/squid directory. Then, we put the pid file of Squid under the /var/run directory, and finally, locate the /icons directory of Squid under /usr/lib/squid/icons with the variable (libexecdir) as shown above.

One important note here is the location of the Squid cache directory. As we can see, we relocate it under the /var/spool/squid directory since the file system (/var/spool) should be on its own partition. This allows us to isolate this file system from the rest of our operating system and to eliminate possible buffer overflow attacks. Also having the directory where the Squid cache will reside on its own partition will allow us to improve performance by tuning parameters of this separate partition with Linux commands like ulimit, etc.

Step 5
Once the modifications have been made to the related Squid source files, it is time configure and optimize Squid for our system.

- To configure and optimize Squid use the following compilation lines:

```
CFLAGS="-O2 -march=i686 -funroll-loops" \
./configure \
--exec_prefix=/usr \
--bindir=/usr/sbin \
--libexecdir=/usr/lib/squid \
--localstatedir=/var \
--sysconfdir=/etc/squid \
--enable-dlmalloc \
--enable-gnuregex \
--enable-xmalloc-statistics \
--with-pthreads \
--enable-removal-policies="heap" \
--enable-storeio=diskd,ufs \
--enable-delay-pools \
--enable-cache-digests \
--enable-err-language=English \
--enable-poll \
--enable-linux-netfilter \
--enable-truncate
```

This tells Squid to set itself up for this particular configuration setup with:

- Link Squid with an external malloc library to improve its cache performance.
- Compile Squid with the GNUregex feature enable.
- Show malloc statistics in status page (cachemgr.cgi).
- Use POSIX Threads to improve Squid performance on Linux.
- Use the heap-replacement feature of Squid to have the choice of various cache replacement algorithms, instead of the standard LRU algorithm for better performance.
- Build support for ufs & diskd I/O modules for better performance.
- Use the delay pools feature of Squid to limit and control bandwidth usage for users.
- Use Squid Cache Digests feature to improve client response time and network utilization.
- Select which default language will be used and installed by Squid for Error pages report.
- Enable poll() instead of select() since it's preferred over select.
- Enable transparent proxy support for Linux kernel 2.4.
- Enable truncate to clean some performance improvements when removing cached files.

Step 6
Now, we must make a list of all existing files on the system before installing the software, and one afterwards, then compare them using the **diff** utility tool of Linux to find out what files are placed where and finally install the Squid Proxy Server:

```
[root@deep squid-2.4.STABLE7]# make all
[root@deep squid-2.4.STABLE7]# cd auth_modules
[root@deep auth_modules]# cd NCSA
[root@deep NCSA]# make
[root@deep NCSA]# cd ../PAM
[root@deep PAM]# make
[root@deep PAM]# cd ../SMB
[root@deep SMB]# make SAMBAPREFIX=/usr
[root@deep SMB]# cd ../getpwnam
[root@deep getpwnam]# make
[root@deep getpwnam]# cd
[root@deep root]# find /* > Squid1
[root@deep root]# cd /var/tmp/squid-2.4.STABLE7/
[root@deep squid-2.4.STABLE7]# make install
[root@deep squid-2.4.STABLE7]# cd auth_modules
[root@deep auth_modules]# install -m 4511 PAM/pam_auth /usr/lib/squid/
[root@deep auth_modules]# install -m 0511 NCSA/ncsa_auth /usr/lib/squid/
[root@deep auth_modules]# install -m 0511 SMB/smb_auth /usr/lib/squid/
[root@deep auth_modules]# install –m 0511 getpwnam/getpwnam_auth
/usr/lib/squid/
[root@deep auth_modules]# mkdir -p /var/spool/squid
[root@deep auth_modules]# mkdir -p /var/log/squid
[root@deep auth_modules]# chown -R squid.squid /var/spool/squid/
[root@deep auth_modules]# chown -R squid.squid /var/log/squid/
[root@deep auth_modules]# chmod 0750 /var/spool/squid/
[root@deep auth_modules]# chmod 0750 /var/log/squid/
[root@deep auth_modules]# rm -rf /var/logs/
[root@deep auth_modules]# rm -f /usr/sbin/RunCache
[root@deep auth_modules]# rm -f /usr/sbin/RunAccel
[root@deep auth_modules]# strip /usr/sbin/squid
[root@deep auth_modules]# strip /usr/sbin/client
[root@deep auth_modules]# strip /usr/lib/squid/*
[root@deep auth_modules]# /sbin/ldconfig
[root@deep auth_modules]# cd
[root@deep root]# find /* > Squid2
[root@deep root]# diff Squid1 Squid2 > Squid-Installed
```

The **make all** command will compile all source files into executable binaries that can be installed, and **make install** will install the binaries and any supporting files into the appropriate locations. Pay special attention to the authenticator module directory of Squid, we move into this directory (auth_modules) and compile all authenticator modules that may be needed with Squid.

Squid authenticator module is required when you want to authorize and authenticate users before allowing them an access to the Internet or the network. Different authenticator modules using different techniques are available with Squid. In our compilation, we compile Squid authenticator modules for PAM, NCSA, SMB, and getpwnam. You don't need to compile all of them but only the one that you want to use or nothing if you are not intending to provide user authentication for Proxy access.

The **mkdir** command will create two new directories named "squid" under /var/spool and /var/log directory.

The **rm** command will remove the /var/logs directory since it has been created to handle the log files for Squid that we have relocated during compile time into the /var/log/squid directory.

The **chown** command will change the owner of the /var/spool/squid and /var/log/squid directories to be owned by the user squid, and the **chmod** command will make the mode of both squid directories (0750/drwxr-x---) for security reasons. This means that only squid owner and group will be able to access these directories and others will not.

Note that we remove the small scripts named "RunCache" and "RunAccel" which take care of starting Squid in either caching mode or accelerator mode, since we use a better script named "squid" located under /etc/init.d directory that takes advantage of Linux system V.

Finally, the **strip** command will reduce the size of the specified binaries for optimum performance.

Step 7
Once we've configured, optimized, compiled, and installed the Squid Proxy Server software, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- This is done by using the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf squid-version/
[root@deep tmp]# rm -f squid-version-src.tar.gz
```

The rm command as used above will remove all the source files we have used to compile and install Squid. It will also remove the Squid compressed archive from the /var/tmp directory.

Configuring Squid

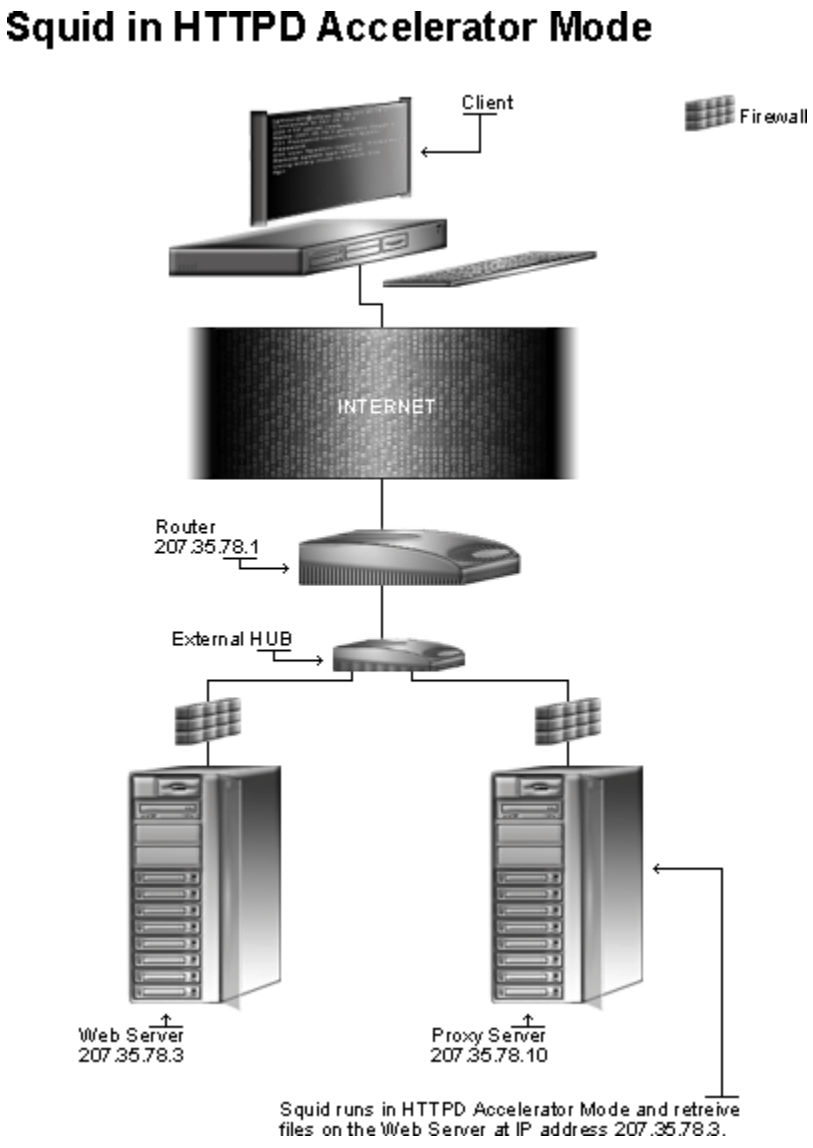
After Squid has been built and installed successfully on your system, your next step is to configure and customize all the required parameters in the different Squid configuration files.

Parameters entered into the Squid configuration file (squid.conf) will decide how the Squid software will run on the server and in which mode (either httpd-accelerator mode or in proxy-caching mode). This shows us that the installation of Squid under Linux does not care and that only the configuration of the squid.conf file will decide whether Squid will run in httpd-accelerator or proxy-caching mode.

- ✓ /etc/squid/squid.conf: (The Squid Configuration File)
- ✓ /etc/sysconfig/squid: (The Squid System Configuration File)
- ✓ /etc/logrotate.d/squid: (The Squid Log Rotation File)
- ✓ /etc/init.d/squid: (The Squid Initialization File)

Running `Squid` in a httpd-accelerator mode

The `squid.conf` file is used to set all the different options for your `Squid` proxy server. In the `Squid` configuration file, we'll configure the `/etc/squid/squid.conf` file to be in httpd-accelerator mode. In this mode, if the Web Server runs on the same server where `Squid` is installed, you must set its daemon to run on port 81. With the `Apache` Web Server, you can do it by changing the line (Port 80) to (Port 81) in its `httpd.conf` file. If the Web Server runs on other servers on your network, like we do, you can keep the same port number (80) for `Apache`, since `Squid` will bind on a different IP number where port (80) is not already in use.



/etc/squid/squid.conf: The Squid Configuration File

The /etc/squid/squid.conf file is the main configuration file for squid. Though there are hundred of option tags in this file, you should only need to change a few options to get Squid up and running. The other options give you amazing flexibility, but you can learn about them once you have Squid running. The text in bold are the parts of the configuration file that must be customized and adjusted to meet our needs. This configuration is suitable when you want to run Squid in httpd-accelerator mode only. Please see later in this chapter for the configuration of Squid in proxy caching mode.

- Edit the **squid.conf** file (vi /etc/squid/squid.conf) and add/change the following options. Below is what we recommend you:

```
http_port 80
icp_port 0
hierarchy_stoplist cgi-bin ?
acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY
cache_mem 128 MB
redirect_rewrites_host_header off
cache_replacement_policy heap GDSF
memory_replacement_policy heap GDSF
cache_dir diskd /var/spool/squid 1000 16 256
cache_store_log none
emulate_httpd_log on
acl all src 0.0.0.0/0.0.0.0
http_access allow all
cache_mgr sysadmin@openna.com
cache_effective_user squid
cache_effective_group squid
httpd_accel_host 207.35.78.3
httpd_accel_port 80
logfile_rotate 0
log_icp_queries off
cachemgr_passwd my-secret-pass all
buffered_logs on
```

This tells the squid.conf file to set itself up for this particular configuration with:

http_port 80
The option "http_port" specifies the port number where Squid will listen for HTTP client requests. If you set this option to port 80, the client will have the illusion of being connected to the Apache Web Server. Since we are running Squid in accelerator mode and our Web Server on other hosts, we must listen on port 80.

icp_port 0
The option "icp_port" specifies the port number where Squid will send and receive ICP requests from neighbouring caches. We must set the value of this option to "0" to disable it, since we are configuring Squid to be in accelerator mode for the Web Server. The ICP feature is needed only in a multi-level cache environment with multiple siblings and parent caches (a feature that only Squid supports compared to other proxy servers on the market). Using ICP in an accelerator mode configuration would add unwanted overhead to Squid. This is an optimization feature.

hierarchy_stoplist cgi-bin ?
The option "hierarchy_stoplist cgi-bin ?" is used to not query neighbor cache for certain objects. The above line is recommended.

```
acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY
```

The options "`acl QUERY urlpath_regex cgi-bin \?`" and "`no_cache deny QUERY`" are used to force certain objects to never be cached, like files under "`cgi-bin`" directory. This is a security feature.

```
cache_mem 128 MB
```

The option "`cache_mem`" specifies the amount of memory (RAM) to be used for caching the so called: In-Transit objects, Hot Objects, Negative-Cached objects. It's important to note that `Squid` can use much more memory than the value you specify in this parameter. For example, if you have 384 MB free for `Squid`, you must put 384/3 = 128 MB here. This is an optimization feature.

```
redirect_rewrites_host_header off
```

The option "`redirect_rewrites_host_header`", if set to "`off`", tells `Squid` to not rewrites any Host: header in redirected requests. It's recommended to set this option to "`off`" if you are running `Squid` in httpd-accelerator mode.

```
cache_replacement_policy heap GDSF
memory_replacement_policy heap GDSF
```

The options "cache_`replacement_policy`" and "`memory_replacement_policy`" specify the cache policy `Squid` will use to determine which objects in the cache must be replaced when the proxy needs to free disk space and which objects are purged from memory when memory space is needed. In our configuration, we choose the `GDSF` (**G**reedy-**D**ual **S**ize **F**requency) policy as our default policy. See http://www.hpl.hp.com/techreports/1999/HPL-1999-69.html and http://fog.hpl.external.hp.com/techreports/98/HPL-98-173.html for more information.

```
cache_dir diskd /var/spool/squid 1000 16 256
```

The option "`cache_dir`" specifies in order: which kind of storage system to use and in our case we choose to use the new `DISKD` storage format of `Squid`, the name of the cache directory (`/var/spool/squid`), the disk space in megabytes to use under this directory (`1000 MB`), the number of first-level subdirectories to be created under the cache directory (`16`), and the number of second-level subdirectories to be created under each first-level cache directory (`256`). In accelerator mode, this option is directly related to the size and number of files that you want to serve with your `Apache` web server. In our example, we suppose that the total size of your web directory will be 1000 MB. Don't forget to change this value to fit the size of your web directory.

```
cache_store_log none
```

The option "`cache_store_log`" logs the activity of the storage manager to the specified file. It shows which objects are ejected from the `Squid` cache, which objects are saved and for how long. We can safely set this option to "`none`" to disable the feature because there are not really any utilities to analyze this data.

```
emulate_httpd_log on
```

The option "`emulate_httpd_log`" if set to "`on`" specifies that `Squid` should emulate the log file format of the `Apache` Web Server. This is very useful if you want to use a third party program like `Webalizer` to analyze and produce static report of the `Squid` Server.

```
acl all src 0.0.0.0/0.0.0.0
http_access allow all
```

The options "`acl`" and "`http_access`" specify and define an access control list to be applied on the `Squid` Proxy Server in httpd-accelerator mode. Our "`acl`" and "`http_access`" option are not restricted, and allows everyone to connect to the proxy server since we use this proxy to accelerate the public `Apache` Web Server. See your `Squid` documentation for more information when using `Squid` in non-httpd-accelerator mode.

```
cache_mgr sysadmin@openna.com
```
The option "cache_mgr" specifies the email-address of the administrator responsible for the Squid Proxy Server. This person is the one who will receive mail if Squid encounter problems. You can specify the name or the complete email address in this option. In our example, we specify the complete email address to be more verbose when errors are encounter.

```
cache_effective_user squid
cache_effective_group squid
```
The options "cache_effective_user" and "cache_effective_group" specify the UID/GID that the cache will run on. Don't forget to never run Squid as "root". In our configuration we use the UID "squid" and the GID "squid" that we have created previously in this chapter. This is a security feature.

```
httpd_accel_host 207.35.78.3
httpd_accel_port 80
```
The options "httpd_accel_host" and "httpd_accel_port" specify to Squid the IP address and port number where the real HTTP Server (i.e. Apache) resides. These are some of the most important parameters when configuring Squid to run in httpd-accelerator mode. In our configuration, the real HTTP Web Server is on IP address 207.35.78.3 (www.openna.com) and on port (80). "www.openna.com" is another FDQN on our network, and since the Squid Proxy Server doesn't reside on the same host where our Apache HTTP Web Server runs, we can use port (80) for our Squid Proxy Server, and port (80) for our Apache Web Server, and the illusion is perfect.

```
logfile_rotate 0
```
The option "logfile_rotate" specifies the number of logfile rotations that we want the Squid program to make. Setting the value to 0 will disable the default rotation and will let us control this feature through our personal logrotate script file. This is what we need to do on Linux since we use our own log script file to make the appropriate rotation of Squid log files.

```
log_icp_queries off
```
The option "log_icp_queries" specifies if you want ICP queries (remember, ICP is used to exchange hints about the existence of URLs in neighbor caches) to be logged to the "access.log" file or not. Since we don't use the ICP feature of Squid in httpd-accelerator mode configuration, we can safely set this option to "off".

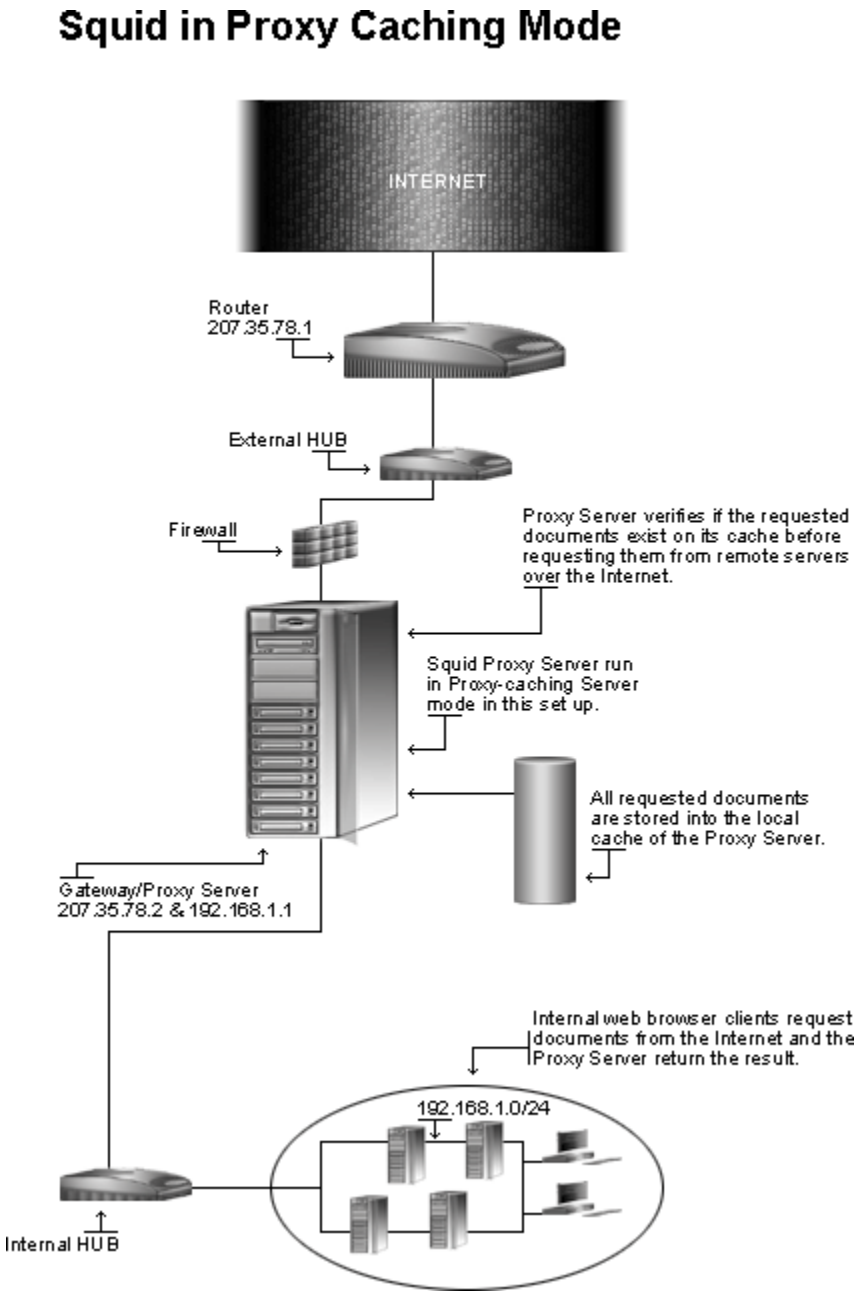
```
cachemgr_passwd my-secret-pass all
```
The option "cachemgr_passwd" specifies a password that will be required for accessing the operations of the "cachemgr.cgi" utility program that comes with Squid. This CGI program is designed to run through a web interface and outputs statistics about the Squid configuration and performance. The <my-secret-pass> is the password that you have chosen, and the keyword <all> specifies to set this password to be the same for all actions you can perform with this program. See "The cachemgr.cgi program utility of Squid", below in this chapter for more information.

```
buffered_logs on
```
The option "buffered_logs", if turned "on", can speed up the writing of some log files slightly. This is an optimization feature.

Running `Squid` in proxy-caching mode

With some minor modifications to the `squid.conf` file we defined earlier to run in httpd-accelerator mode, we can run `Squid` as a proxy-caching server. With a proxy-caching server, all users in your corporate network will use `Squid` to access the Internet. This is the configuration that you must use for a Gateway Server running `Squid` and it is the most commonly used configuration by Linux administrators who install `Squid` on their servers.

With this configuration, you can have complete control, apply special policies on what can be viewed, accessed, and downloaded. You can also control bandwidth usage, connection time, and so on. A proxy caching server can be configured to run as stand-alone server for your corporation, or to use and share caches hierarchically with other proxy servers around the Internet.



/etc/squid/squid.conf: The Squid Configuration File

To set up Squid as a proxy-caching server, we use the same configuration file as before but with some additional modifications to the default in relation to Squid in httpd-accelerator mode. The text in bold are the parts of the configuration file that must be customized and adjusted to satisfy our needs.

The rest of the parameters are the same as for Squid in httpd-accelerator mode and I recommend you to read the configuration section related to Squid in httpd-accelerator mode for more information on each option. This configuration is suitable when you want to run Squid in proxy-caching mode only. Please see the information earlier in this chapter for the configuration of Squid in httpd-accelerator mode.

- Edit the **squid.conf** file (vi /etc/squid/squid.conf) and add/change the following options for Squid in proxy caching mode that run as a stand-alone server. Below is what we recommend:

```
icp_port 0
hierarchy_stoplist cgi-bin ?
acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY
cache_mem 128 MB
cache_replacement_policy heap GDSF
memory_replacement_policy heap GDSF
cache_dir diskd /var/spool/squid 2000 16 256
cache_store_log none
acl localnet src 192.168.1.0/255.255.255.0
acl localhost src 127.0.0.1/255.255.255.255
acl Safe_ports port 80 443 210 70 21 1025-65535
acl CONNECT method CONNECT
acl all src 0.0.0.0/0.0.0.0
http_access allow localnet
http_access allow localhost
http_access deny !Safe_ports
http_access deny CONNECT
http_access deny all
cache_mgr sysadmin@openna.com
cache_effective_user squid
cache_effective_group squid
logfile_rotate 0
log_icp_queries off
cachemgr_passwd my-secret-pass all
buffered_logs on
```

NOTE: In the above configuration example, the default Proxy port '3128' will be used. If you prefer to use another port like '8080', all you will have to do will be to add the parameter "http_port 8080" and configure your clients accordingly.

One of the big differences with the Squid httpd-accelerator mode configuration file is the use of **A**ccess **C**ontrol **L**ists (ACL). For Squid in Proxy-Caching mode, this feature allows you to restrict access based on source IP address (src), destination IP address (dst), source domain, destination domain, time, and so on. Many types exist with this feature, and you should consult the "squid.conf" file for a complete list.

The four most commonly used types are as follows:

```
acl  name      type       data
 |     |         |         |
acl some-name src a.b.c.d/e.f.g.h       # ACL restrict access based on source IP address
acl some-name dst a.b.c.d/e.f.g.h       # ACL restrict access based on destination IP address
acl some-name srcdomain foo.com         # ACL restrict access based on source domain
acl some-name dstdomain foo.com         # ACL restrict access based on destination domain
```

For example, to restrict access to your `Squid` proxy server to only your internal clients, and to a specific range of designated ports, something like the following will do the job:

```
# Our ACL Elements
acl localnet src 192.168.1.0/255.255.255.0
acl localhost src 127.0.0.1/255.255.255.255
acl Safe_ports port 80 443 210 70 21 1025-65535
acl CONNECT method CONNECT
acl all src 0.0.0.0/0.0.0.0

# Our Access Lists
http_access allow localnet
http_access allow localhost
http_access deny !Safe_ports
http_access deny CONNECT
http_access deny all
```

Let's explain what's going on. First we can see that there are two distinct groups `acl` and `http_access`; all the '`acl`' parts with their different types are called "ACL elements" and all the '`http_access`' parts with their different types are called "Access Lists". We use "ACL elements" to define our names, source IP addresses, destination IP addresses, source domain, destination domain, port, etc and "Access Lists" to define the action that must be associated with the "ACL elements". The action can be to deny or allow the "ACL elements" rules.

In our example above, we define five "ACL elements":

```
acl localnet src 192.168.1.0/255.255.255.0
acl localhost src 127.0.0.1/255.255.255.255
acl Safe_ports port 80 443 210 70 21 1025-65535
acl CONNECT method CONNECT
acl all src 0.0.0.0/0.0.0.0
```

and five "Access Lists" pertaining to the "ACL elements":

```
http_access allow localnet
http_access allow localhost
http_access deny !Safe_ports
http_access deny CONNECT
http_access deny all
```

The Squid program reads the access lines in the order that there are appearing. Pertaining to our example, Squid will interpret all the access lines as follow:

```
1)  Read → acl localnet src 192.168.1.0/255.255.255.0
2)  Read → acl localhost src 127.0.0.1/255.255.255.255
3)  Read → acl Safe_ports port 80 443 210 70 21 1025-65535
4)  Read → acl CONNECT method CONNECT
5)  Read → acl all src 0.0.0.0/0.0.0.0
6)  Apply → http_access allow localnet
7)  Apply → http_access allow localhost
8)  Apply → http_access deny !Safe_ports
9)  Apply → http_access deny CONNECT
10) Apply → http_access deny all
```

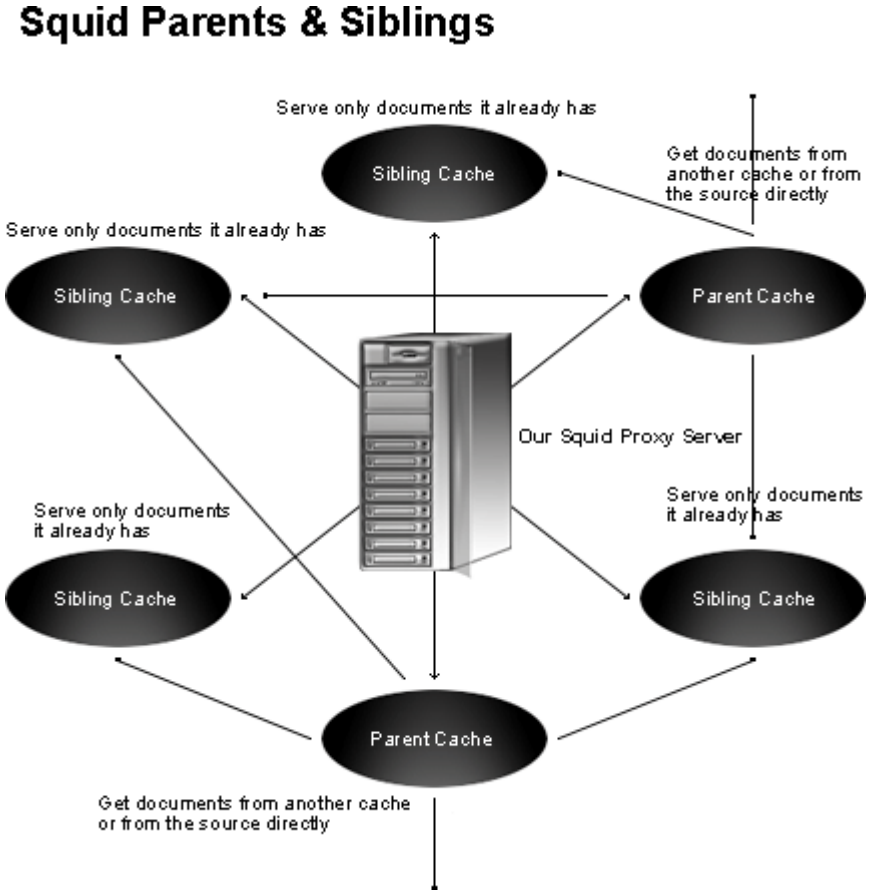
This ACL configuration will allow all internal clients from the private class C 192.168.1.0 to access the proxy server; it's also recommended that you allow the localhost IP (a special IP address used by your own server) to access the proxy.

After we choose a range of ports (80=http, 443=https, 210=wais, 70=gopher, and 21=ftp) which our internal clients can use to access the Internet, we deny the CONNECT method to prevent outside people from trying to connect to the proxy server, and finally, we deny all source IP address and ports on the proxy server.

Multi-level Web Caching

The second method of proxy caching is the so-called "Multi-level Web Caching" where you choose to share and cooperate with more proxy-cache servers on the Internet. With this method, your organization uses the cache of many others proxy cache servers, and to compensate, the other cache server can use yours.

It's important to note that in this situation, the proxy cache can play two different roles in the hierarchy. It can be configured as a **sibling** cache, and be able to only serve documents it already has, or it can be configured as a **parent** cache, and be able to get documents from another cache or from the source directly.



NOTE: A good strategy to avoid generating more network traffic than without web caching is to choose to have several **sibling** caches and only a small number of **parent** caches.

`/etc/sysconfig/squid`: The `Squid` System Configuration File

The `/etc/sysconfig/squid` file is used to specify `Squid` system configuration information, such as if `Squid` should enable initial `DNS` checks at start-up, and the value of time to wait for `Squid` to shut down when asked.

- Create the **`squid`** file (`touch /etc/sysconfig/squid`) and add the following lines:

```
# If you most likely will not to have an Internet connection when you
# start Squid, uncomment this. The -D option disables initial dns checks.
#SQUID_OPTS="-D"

# Time to wait for Squid to shut down when asked. Should not be necessary
# most of the time.
SQUID_SHUTDOWN_TIMEOUT=100
```

/etc/logrotate.d/squid: The Squid Log Rotation Configuration File

The /etc/logrotate.d/squid file is responsible for rotating log files related to Squid software automatically each week via syslog. If you are not familiar with syslog, look at the syslog.conf (5) manual page for a description of the syslog configuration file, or the syslogd (8) manual page for a description of the syslogd daemon.

- Create the **squid** file (touch /etc/logrotate.d/squid) and add the following lines:

```
/var/log/squid/access.log {
    weekly
    rotate 5
    copytruncate
    compress
    notifempty
    missingok
}
/var/log/squid/cache.log {
    weekly
    rotate 5
    copytruncate
    compress
    notifempty
    missingok
}

/var/log/squid/store.log {
    weekly
    rotate 5
    copytruncate
    compress
    notifempty
    missingok
# This script asks Squid to rotate its logs on its own. Restarting Squid
# is a long process and it is not worth doing it just to rotate logs.
    postrotate
      /usr/sbin/squid -k rotate
    endscript
}
```

/etc/init.d/squid: The Squid Initialization File

The /etc/init.d/squid script file is responsible for automatically stopping and starting the Squid Internet Object Cache on your server. Loading the squid daemon, as a standalone daemon will eliminate load time and will even reduce swapping, since non-library code will be shared. Please note that the following script is suitable for Linux operating systems that use SystemV. If your system uses some other method like BSD, you'll have to adjust the script below to make it work for you.

Step 1
Create the **squid** script file (touch /etc/init.d/squid) and add the following lines:

```
#!/bin/bash

# This shell script takes care of starting and stopping Squid (Proxy server).
#
# chkconfig: 345 90 25
# description: Squid - Internet Object Cache. Internet object caching is \
#       a way to store requested Internet objects (i.e., data available \
```

```
#       via the HTTP, FTP, and gopher protocols) on a system closer to the \
#       requesting site than to the source. Web browsers can then use the \
#       local Squid cache as a proxy HTTP server, reducing access time as \
#       well as bandwidth consumption.
#
# processname: squid

# pidfile: /var/run/squid.pid
# config: /etc/squid/squid.conf

PATH=/usr/bin:/sbin:/bin:/usr/sbin
export PATH

# Source function library.
. /etc/rc.d/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

# Check if the squid.conf file is present.
[ -f /etc/squid/squid.conf ] || exit 0

# Source Squid configureation.
if [ -f /etc/sysconfig/squid ]; then
        . /etc/sysconfig/squid
else
  SQUID_OPTS="-D"
  SQUID_SHUTDOWN_TIMEOUT=100
fi

# Determine the name of the squid binary.
[ -f /usr/sbin/squid ] && SQUID=squid
[ -z "$SQUID" ] && exit 0

prog="$SQUID"

# Determine which one is the cache_swap directory
CACHE_SWAP=`sed -e 's/#.*//g' /etc/squid/squid.conf | \
        grep cache_dir |  awk '{ print $3 }'`
[ -z "$CACHE_SWAP" ] && CACHE_SWAP=/var/spool/squid

RETVAL=0

start() {
    for adir in $CACHE_SWAP; do
        if [ ! -d $adir/00 ]; then
             echo -n "init_cache_dir $adir... "
             $SQUID -z -F 2>/dev/null
        fi
    done
    echo -n $"Starting $prog: "
    $SQUID $SQUID_OPTS 2> /dev/null &
    # Trap and prevent certain signals from being sent to the Squid process.
    trap '' 1 2 3 18
    RETVAL=$?
    [ $RETVAL -eq 0 ] && touch /var/lock/subsys/$SQUID
    [ $RETVAL -eq 0 ] && echo_success
    [ $RETVAL -ne 0 ] && echo_failure
    echo
    return $RETVAL
```

```
}

stop() {
    echo -n  $"Stopping $prog: "
    $SQUID -k check >/dev/null 2>&1
    RETVAL=$?
    if [ $RETVAL -eq 0 ] ; then
        $SQUID -k shutdown &
        rm -f /var/lock/subsys/$SQUID
        timeout=0
        while : ; do
                [ -f /var/run/squid.pid ] || break
                if [ $timeout -ge $SQUID_SHUTDOWN_TIMEOUT ]; then
                    echo
                    return 1
                fi
                sleep 2 && echo -n "."
                timeout=$((timeout+2))
        done
        echo_success
        echo
    else
        echo_failure
        echo
    fi
    return $RETVAL
}

reload() {
        $SQUID $SQUID_OPTS -k reconfigure
}

restart() {
        stop
        start
}

condrestart() {
        [ -e /var/lock/subsys/squid ] && restart || :
}

# See how we were called.
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  reload)
        reload
        ;;
  restart)
        restart
        ;;
  condrestart)
         condrestart
        ;;
        *)
        echo $"Usage: $0 {start|stop|reload|restart|condrestart}"
        exit 1
esac
exit $?
```

Step 2
Once the /etc/init.d/squid script file has been created, it is important to make it executable, change its default permissions, create the necessary links and start it. Making this file executable will allow the system to run it, changing its default permission is to allow only the root user to change this file for security reason, and creation of the symbolic links will let the process control initialization of Linux which is in charge of starting all the normal and authorized processes that need to run at boot time on your system to start the program automatically for you at each system reboot.

- To make this script executable and to change its default permissions, use the commands:
```
[root@deep /]# chmod 700 /etc/init.d/squid
[root@deep /]# chown 0.0 /etc/init.d/squid
```

- To create the symbolic rc.d links for Squid, use the following commands:
```
[root@deep /]# chkconfig --add squid
[root@deep /]# chkconfig --level 345 squid on
```

- To start Squid software manually, use the following command:
```
[root@deep /]# /etc/init.d/squid start
Starting squid:          [OK]
```

Running Squid with Users Authentication Support

Squid is a complete Proxy software solution that provides many useful features for the administrator and it is up to us to use and configure them. In this section, we'll discuss the Proxy Authentication mechanism that provides a way to authenticate internal users who can use it to access the Internet. This allows us to add an additional layer of security on which users need to have rights (the authorization) to access the Internet via the Gateway server in the enterprise.

The Squid source code comes with a few authentication processes. These include:

LDAP: Uses the **L**ightweight **D**irectory **A**ccess **P**rotocol.
NCSA: Uses an NCSA-style username and password file.
MSNT: Uses a Windows NT authentication domain.
PAM: Uses the Linux **P**luggable **A**uthentication **M**odules scheme.
SMB: Uses a SMB server like Windows NT or Samba.
getpwam: Uses the old-fashioned Unix password file.

In order to authenticate users, you need to compile and install one of the above supplied authentication modules. In our compilation of Squid, we have already included the most interesting authentication modules, which were NCSA, PAM, SMB, and getpwam.

One problem with all of these authentication modules is the fact that the supplied username and password are essentially sent in clear text between the browser and the proxy. Therefore, administrators should not set-up the same username and password that users would use for account login on the server (if they are allowed) or for email accounts.

This means that we have to create a null account, with no valid shell, no files owned-nothing but a UID and a GID for every user that will use the Squid Proxy Server, with authentication, to access the Internet. The best authentication module to accomplish this will be the PAM authentication module because it will allow us to manage proxy users' authentication access through the /etc/passwd file in the easiest and fastest manner available. It would also allow us to create the null account without problem. Below, we will show you, how to use and configure the PAM authentication module with Squid.

Step 1
The first step in our procedure will be to create a PAM configured authentication service called "squid" under the /etc/pam.d directory to allow us to authenticate Squid users.

- Create the **squid** file (touch /etc/pam.d/squid) and add the following lines:

```
#%PAM-1.0
auth       required     /lib/security/pam_stack.so service=system-auth
account    required     /lib/security/pam_stack.so service=system-auth
```

Step 2
Now, it is time to let Squid know which authentication program to use in squid.conf. In our case, we have to tell it to use the PAM authentication module.

- Edit the **squid.conf** file (vi /etc/squid/squid.conf) and add the following line. Text in bold is what we have added to our default Squid example configuration file. Below is what we recommend you enter:

```
icp_port 0
hierarchy_stoplist cgi-bin ?
acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY
cache_mem 128 MB
cache_replacement_policy heap GDSF
memory_replacement_policy heap GDSF
cache_dir diskd /var/spool/squid 2000 16 256
cache_store_log none
authenticate_program /usr/lib/squid/pam_auth /etc/passwd
acl localnet src 192.168.1.0/255.255.255.0
acl localhost src 127.0.0.1/255.255.255.255
acl Safe_ports port 80 443 210 70 21 1025-65535
acl CONNECT method CONNECT
acl all src 0.0.0.0/0.0.0.0
http_access allow localnet
http_access allow localhost
http_access deny !Safe_ports
http_access deny CONNECT
http_access deny all
cache_mgr sysadmin@openna.com
cache_effective_user squid
cache_effective_group squid
logfile_rotate 0
log_icp_queries off
cachemgr_passwd my-secret-pass all
buffered_logs on
```

In the above line, we specify the name of the program (pam_auth) to use for user authentication, plus any command line options if necessary (/etc/passwd).

Step 3
Next, we have to add some `proxy_auth ACL` entries to our `squid.conf` configuration file to control and authorize the access.

- Edit the **`squid.conf`** file (`vi /etc/squid/squid.conf`) and add the following options to the `squid.conf` file to be able to authenticate and control users access. Again the text in bold is what we have added to the previous `Squid` example configuration file in step 2. Below is what we recommend you enter:

```
icp_port 0
hierarchy_stoplist cgi-bin ?
acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY
cache_mem 128 MB
cache_replacement_policy heap GDSF
memory_replacement_policy heap GDSF
cache_dir diskd /var/spool/squid 2000 16 256
cache_store_log none
authenticate_program /usr/lib/squid/pam_auth /etc/passwd
acl authenticated proxy_auth REQUIRED
acl localnet src 192.168.1.0/255.255.255.0
acl localhost src 127.0.0.1/255.255.255.255
acl Safe_ports port 80 443 210 70 21 1025-65535
acl CONNECT method CONNECT
acl all src 0.0.0.0/0.0.0.0
http_access allow authenticated
http_access allow localnet
http_access allow localhost
http_access deny !Safe_ports
http_access deny CONNECT
http_access deny all
cache_mgr sysadmin@openna.com
cache_effective_user squid
cache_effective_group squid
logfile_rotate 0
log_icp_queries off
cachemgr_passwd my-secret-pass all
buffered_logs on
```

The added lines mean that any authenticated user will match the `ACL` named "`authenticated`". The string `REQUIRED` is used to accept any valid username.

> **NOTE:** Don't forget to restart your `Squid` Proxy Server for the changes to take effect. The order in which each line appears in the `Squid` configuration file is important and you have to respect them. You can't just add '`acl`' or '`http_access`' parameters, wherever you want. Because the program reads and interprets each access line in the order that they appear. The above configurations CAN'T be used in conjunction with the `ACL` configuration for **Banning all Destination addresses except one** (see further down in this chapter for more information).

Step 4
One of the last steps is to create accounts for all users who will be allowed to access the Internet with `Squid` after proper authentication with a username and password. Remember, we have to create null account, with no valid shell for our users.

- To create null user account, use the following command:
 `[root@deep /]# useradd -s /bin/false gmourani`

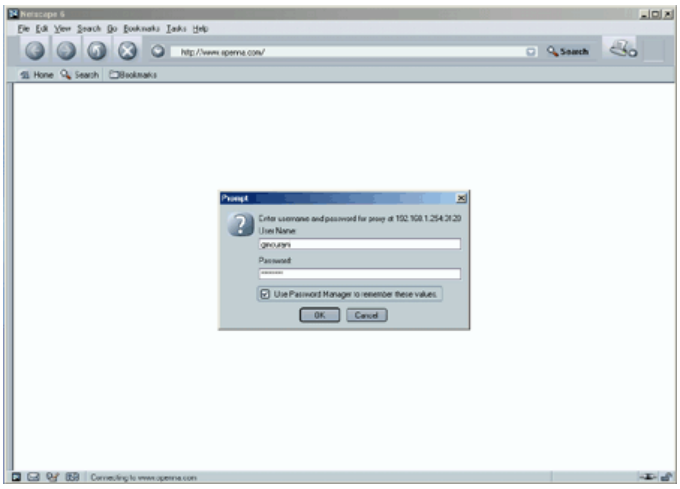
The above command will create a null account, with no password, no valid shell, no files owned- nothing but a `UID` and a `GID`.

- To set a password for this new user, use the following command:

```
[root@deep /]# passwd gmourani
Changing password for user gmourani
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully
```

NOTE: It is NOT important to create a home directory for the users (i.e. `/home/gmourani`). `Squid` with Users Authentication Support can run even if home directories are not created for the users. All we need for authentication is a username and password. Therefore, a home directory is futile and since we do not give shell access, there is really no reason for users to have a home directory on the system.

Step 5
Finally, open your favorite browser and enter the username and password to access the Internet with `Squid` as your Proxy Caching Gateway Server.



Securing Squid

This section deals specifically with actions we can take to improve and tighten security under Squid. As with the other chapters, the interesting points here are that we refer to the features available within the base installed program and not any additional software.

More control on mounting the cache directory of Squid

If you have created the cache directory of Squid in a separate partition your system (i.e. /var/spool), like we have done during the initial set-up of Linux, then you can use the noexec, nodev, and nosuid features of Linux to improve and consolidate the cache security.

These features can be set up in the /etc/fstab file to inform the system to not allow execution of any binaries (noexec), to not interpret character or block special devices (nodev), and to not allow set-user-identifier or set-group-identifier bits to take effect (nosuid) on the mounted file system (/var/spool in our example).

Applying this procedure on the partition where the Squid Cache resides will help to eliminate the possibility of DEV, SUID/SGID, and execution of any binaries that may be in the Squid cache.

Step 1

- Edit the **fstab** file (vi /etc/fstab) and add in the line that refer to /var/spool file system the following options after the defaults option as show below:

```
LABEL=/var/spool  /var/spool  ext3    defaults,noexec,nodev,nosuid    1 2
```

Step 2

Once you have made the necessary adjustments to the /etc/fstab file, it is time to inform the system about the modification.

- This can be accomplished with the following commands:

```
[root@deep /]# mount /var/spool -oremount
```

Each file system that has been modified must be remounted with the command as shown previously. In our example we have modified the /var/spool file system and it is for this reason that we remount this file system with the above command.

> **NOTE:** If /var/spool is not a file system but just a directory, then the above command obviously will not work. The '-oremount' option of the Linux 'mount' command is used to remount a file system, which resides on its own partition on your computer.

Step 3

- You can verify if the modifications have been correctly applied to the Linux system with the following command:

```
[root@deep /]# cat /proc/mounts
/dev/root /   ext2        rw 0 0
/proc /proc   proc        rw 0 0
/dev/sda1     /boot       ext3    rw 0 0
/dev/sda9     /chroot     ext3    rw 0 0
/dev/sda8     /home       ext3    rw 0 0
/dev/sda13    /tmp        ext3    rw 0 0
/dev/sda7     /usr        ext3    rw 0 0
/dev/sda11    /var        ext3    rw 0 0
/dev/sda12    /var/spool  ext3    rw,noexec,nodev,nosuid 0 0
none  /dev/pts    devpts  rw 0 0
```

This command will show you all file system in your Linux server with parameters applied to them. If you see something like the following, congratulations!

```
/var/spool /var/spool ext3    rw,noexec,nodev,nosuid 0 0
```

Immunize the `Squid` configuration file

As we already know, the immutable bit can be used to prevent deletion, overwriting, or creation of a symbolic link to a file. Once your `squid.conf` file has been configured, it's a good idea to immunize it with the following command:

```
[root@deep /]# chattr +i /etc/squid/squid.conf
```

Banning all Destination addresses except one

In the university libraries, we can often see computers available to students that want to, for their studies, search for a specific author. Administrators have the task of configuring the computer to allow only searches on one site where the database of all books and authors are stored. Therefore, they don't want to give students access to other sites on the Internet but just to the database site.

With Squid as the Proxy Server, this can be accomplished easily by adding the right `ACL` to its existing configuration file. In the next example, we introduce new `ACL` rules to our `Squid` example configuration file to do just this.

- Edit the **`squid.conf`** file (`vi /etc/squid/squid.conf`) and add/change the following options. Text in bold are what we have added/changed to our default `Squid` example configuration file. Below is what we recommend you enter:

```
icp_port 0
hierarchy_stoplist cgi-bin ?
acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY
cache_mem 128 MB
cache_replacement_policy heap GDSF
memory_replacement_policy heap GDSF
cache_dir diskd /var/spool/squid 2000 16 256
cache_store_log none
acl localhost src 127.0.0.1/255.255.255.255
acl Safe_ports port 80 443 1025-65535
acl CONNECT method CONNECT
acl DATABASE dst 207.78.0.1
```

```
acl INTERNET dst 0.0.0.0/0.0.0.0
acl all src 0.0.0.0/0.0.0.0
http_access allow localhost
http_access allow DATABASE
http_access deny !Safe_ports
http_access deny CONNECT
http_access deny INTERNET
http_access deny all
cache_mgr sysadmin@openna.com
cache_effective_user squid
cache_effective_group squid
logfile_rotate 0
log_icp_queries off
cachemgr_passwd my-secret-pass all
buffered_logs on
```

This new ACL configuration allows the localhost and any internal clients to access the Proxy Server on the standard ports HTTP, HTTPS and all non-privileged ports, only when they want to connect to the destination IP address (207.78.0.1), which runs our database site. In this way, we limit web access to only one site and students cannot access the Internet.

> **NOTE:** Don't forget to restart your Squid Proxy Server for the changes to take effect. The order in which each line appears in the Squid configuration file is important and you have to respect them. You can't just add 'acl' or 'http_access' parameters, wherever you want. The program reads and interprets each access line in the order that they appear. The above configurations CAN'T be used in conjunction with the ACL configuration for **Users Authentication Support** (see further up in this chapter for more information).

Allowing access to the Internet at specific times

Let's say you want all of your internal hosts only be allowed access to the Internet during working hours (8:30 - 17:30). You can use something like this.

- Edit the **squid.conf** file (vi /etc/squid/squid.conf) and add the following options. Text in bold is what we have added to our default Squid example configuration file:

```
icp_port 0
hierarchy_stoplist cgi-bin ?
acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY
cache_mem 128 MB
cache_replacement_policy heap GDSF
memory_replacement_policy heap GDSF
cache_dir diskd /var/spool/squid 2000 16 256
cache_store_log none
acl staff src 192.168.1.0/255.255.255.0
acl localhost src 127.0.0.1/255.255.255.255
acl Safe_ports port 80 443 210 70 21 1025-65535
acl CONNECT method CONNECT
acl WORKING time MTWHF 08:30-17:30
acl all src 0.0.0.0/0.0.0.0
http_access allow staff WORKING
http_access allow localhost
http_access deny !Safe_ports
http_access deny CONNECT
http_access deny staff
```

```
http_access deny all
cache_mgr sysadmin@openna.com
cache_effective_user squid
cache_effective_group squid
logfile_rotate 0
log_icp_queries off
cachemgr_passwd my-secret-pass all
buffered_logs on
```

This new ACL configuration allows all internal clients from the private class C 192.168.1.0 to access the Internet between 08:30 and 17:30. In this way, we limit the time when our staff can connect to the Internet to the working hours of the company only.

> **NOTE:** Don't forget to restart your Squid Proxy Server for the changes to take effect. The order in which each line appears in the Squid configuration file is important and you have to respect them. You can't just add 'acl' or 'http_access' parameters, wherever you want. Because the program reads and interprets each access line in the order that they appear.

Optimizing Squid

This section deals specifically with the actions we can take to improve and tighten the performance of Squid. Note that we refer to the features available within the base installed program only.

The atime and noatime attributes

The atime and noatime attributes of Linux can be used to get a measurable performance gain in the Squid cache directory (/var/spool/squid). See the chapter related to the kernel in this book for more information on this issue.

Physical memory

The most important resource for Squid is physical memory. Your processor does not need to be ultra-fast. Your disk system will be the major bottleneck, so fast disks are also important for high-volume caches. Therefore, our recommendation is to use a SCSI disk with at least 512 MB of physical memory.

Squid Administrative Tools

Now you've configured Squid, tightened its security and optimized it for maximum performance, we can start to play with its utilities.

Stopping Squid process immediately

There are some interesting command line options, especially when we want to stop Squid on the server. Unlike other services that run as daemons on the system, Squid cannot be stopped directly and we have to wait for existing connections to terminate before Squid shutdowns. Sometimes this is not appropriate and we have to stop Squid immediately. This is possible with the following command:

- To stop Squid process immediately, use the following command:
 `[root@deep /]# /usr/sbin/squid -k kill`

This command sends a KILL signal, which causes the Squid process to exit immediately, without closing any connections or log files.

Purging object from your cache

Sometimes when information stored in the Squid cache become inaccurate for some reason or when there's a problem relating to the cached files, it is nice to have an option which purges the cache. We cannot just delete the cache directories and then expect that everything will be fine. There is a command to do it and we must use it.

Step 1
By default, Squid does not allow you to purge objects unless it is configured with access controls in squid.conf. Below, we'll show you the procedure to accomplish this action.

- Edit the **squid.conf** file (vi /etc/squid/squid.conf) and add the following options to the squid.conf file so we can purge objects. The text in bold are what we have added to our default Squid example configuration file. Below is what we recommend you put in your file:

```
icp_port 0
hierarchy_stoplist cgi-bin ?
acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY
cache_mem 128 MB
cache_replacement_policy heap GDSF
memory_replacement_policy heap GDSF
cache_dir diskd /var/spool/squid 2000 16 256
cache_store_log none
acl localnet src 192.168.1.0/255.255.255.0
acl localhost src 127.0.0.1/255.255.255.255
acl Safe_ports port 80 443 210 70 21 1025-65535
acl CONNECT method CONNECT
acl PURGE method PURGE
acl all src 0.0.0.0/0.0.0.0
http_access allow localnet
http_access allow localhost
http_access allow PURGE localhost
http_access deny !Safe_ports
http_access deny CONNECT
http_access deny PURGE
http_access deny all
cache_mgr sysadmin@openna.com
cache_effective_user squid
cache_effective_group squid
logfile_rotate 0
log_icp_queries off
cachemgr_passwd my-secret-pass all
buffered_logs on
```

This new ACL configuration allows only purge requests of the cache if the request is made from the localhost (on the terminal of your Gateway Server), and denies all other purge requests.

> **NOTE:** Don't forget to restart your Squid Proxy Server for the changes to take effect. The order in which each line appears in the Squid configuration file is important and you have to respect that. You can't just add 'acl' or 'http_access' parameters, wherever you want. Because the program reads and interprets each access line in the order that they appears.

Step 2
Once the correct ACL have been added to your squid.conf file to allow purge requests on the Proxy Server, we have to use the client program that comes with Squid to purge an object.

- This procedure can be accomplished with the following commands:
```
[root@deep /]# client -m PURGE http://www.mydomain.com/
```

Where <www.mydomain.com> is the object that you want to purge. If the purge was successful, you will see a "200 OK" response. If the object was not found in the cache, you will see a "404 Not Found" response.

NOTE: The PURGE feature of Squid works only when Users Authentication Support is disabled in the Squid configuration file. The client program of Squid is not capable of using User Authentication because it doesn't have the option to specify a username or password through its command line.

The cachemgr.cgi program utility of Squid

The cachemgr.cgi utility program, which is available by default when you compile and install Squid on your system, is designed to run through a web interface, and outputs various statistics about Squid configuration and performance.

Personally, I don't recommend you use it. The cachemgr.cgi is a buggy utility, which provides incomprehensible and cryptic results. Connection to its web interface is not always guaranteed even if you have the proper configuration. I think that more development and a complete revision of its functionality is required. Especially when we want to make a remote connection to its web interface. If you really want to use it, then here are the correct steps you must follow.

This program, by default, is located under the /usr/lib/squid directory, and you have to put it in your "cgi-bin" directory (eg, /home/httpd/cgi-bin) to be able to use it. Follow the simple steps below to use this program.

Step 1
The first step will be to move the "cachemgr.cgi" CGI file from the /usr/lib/squid directory to your /home/httpd/cgi-bin directory.

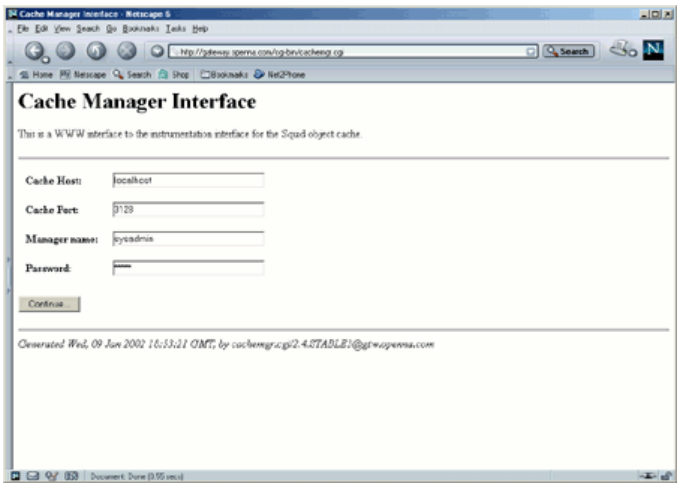
- This procedure can be accomplished with the following command:
```
[root@deep /]# mv /usr/lib/squid/cachemgr.cgi /home/httpd/cgi-bin/
```

Step 2
Once you've put the "cachemgr.cgi" program into your /cgi-bin directory, it is time to change its default mode permission and owner.

- These procedures can be accomplished with the following commands:
```
[root@deep /]# cd /home/httpd/cgi-bin/
[root@deep cgi-bin]# chown 0.0 cachemgr.cgi
[root@deep cgi-bin]# chmod 0511 cachemgr.cgi
```

Step 3
Finally, you can point your web browser to the following address (http://my-web-server/cgi-bin/cachemgr.cgi) to use the various features of this program.

The `<my-web-server>` is the address where your `Apache` web server lives, and `<cachemgr.cgi>` is the `Squid` utility program we have just placed in our "`cgi-bin`" directory to display information and the configuration of our `Squid` Proxy Linux server.



If you have configured the `squid.conf` file to use password authentication for `cachemgr.cgi` (as we do), you'll be asked to enter the "Cache Host", "Cache Port", "Manager Name", and "Password information" before you are able to access the `cachemgr.cgi` program. See the configuration of the `/etc/squid/squid.conf` file, shown earlier, for more information.

> **`WARNING:`** Please note that only a browser running on the `Squid` machine (the Gateway Server) that doesn't use the proxy will be able to connect to the `cachemgr.cgi` web interface. If you try to access the web interface remotely via another system, then the authentication will fail.

CHAPTER 12

SquidGuard Filter

IN THIS CHAPTER

1. Compiling - Optimizing & Installing `SquidGuard`
2. Configuring `SquidGuard`
3. Testing `SquidGuard`
4. Optimizing `SquidGuard`

Linux SquidGuard

Abstract

As we saw in the previous chapter, the Squid ACL (**A**ccess **C**ontrol **L**ists) has some limitations in its functionality and it can become very hard to configure a complex ACL. We need to find another way to simplify the procedure of configuring our ACL and this is possible with plug-in software called SquidGuard.

SquidGuard is a combined filter, redirector and access controller plug-in for Squid. It allows us to improve, in many ways, the default ACL of Squid. We can use it to limit web access for users to a list of accepted/well known web servers and/or URLs like Squid does already but in an easier manner. We can use it to block access to particular listed or blacklisted web servers and/or URLs, block access to URLs matching a list of regular expressions or words, redirect blocked URLs to an "intelligent" CGI based info page, have different access rules based on time of day, day of the week, date etc, and much more.

In general it is a good addition to run with Squid Proxy Server on your Gateway Server for additional security and power. In this chapter, we will show you how to install and configure it to block undesirable websites like porn sites, warez, etc and how to configure it to allow Internet access on specific days and times from our corporate network. We will also merge it with the Squid default ACL to get maximum efficiency and security.

Thousands, even millions, of IP addresses, and URL's can be added to different filters files without sacrificing too much performance of the Squid Proxy Server. This is possible since SquidGuard uses good programming techniques to achieve this, and it is far ahead of its competitors for speed.

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest SquidGuard version number is 1.2.0

The procedures given in this chapter are likely to work on all Linux platforms, but we have only tested it on OpenNA Linux and Red Hat Linux.

Packages

The following is based on information listed by SquidGuard as of 2001/12/18. Please check http://www.squidguard.org/ regularly for the latest status. We chose to install from source because it provides the facility to fine tune the installation.

Source code is available from:
SquidGuard Homepage: http://www.squidguard.org/
SquidGuard FTP Site: 195.70.164.135
You must be sure to download: squidGuard-1.2.0.tar.gz

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all the files installed onto the system if you want to update the package in the future. To solve this problem, it's a good idea to make a list of files on the system before you install SquidGuard, and then one afterwards, and then compare them using the **diff** utility to find out what files were placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep root]# find /* > SquidGuard1
  ```

- And the following one after you install the software:
  ```
  [root@deep root]# find /* > SquidGuard2
  ```

- Then use this command to get a list of what changed:
  ```
  [root@deep root]# diff SquidGuard1 SquidGuard2 > SquidGuard-Installed
  ```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. In our example above, we use the /root directory of the system to store all the generated file lists.

Compiling - Optimizing & Installing SquidGuard

Below are the steps that you must take to configure, compile and optimize the SquidGuard software before installing it onto your system.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- These procedures can be accomplished with the following commands:
  ```
  [root@deep /]# cp squidGuard-version.tar.gz /var/tmp/
  [root@deep /]# cd /var/tmp/
  [root@deep tmp]# tar xzpf squidGuard-version.tar.gz
  ```

Step 2
After that, move into the newly created SquidGuard source directory and perform the following steps to configure and optimize the software for your system.

- To move into the newly created SquidGuard source directory use the command:
  ```
  [root@deep tmp]# cd squidGuard-1.2.0/
  ```

- To configure and optimize SquidGuard use the following compilation lines:
  ```
  CFLAGS="-O2 -march=i686 -funroll-loops" \
  ./configure \
  --prefix=/usr \
  --sysconfdir=/etc \
  --localstatedir=/var \
  --with-sg-config=/etc/squid/squidGuard.conf \
  --with-sg-logdir=/var/log/squid/squidGuard \
  --with-sg-dbhome=/var/spool/squid/squidGuard \
  --with-db-inc=/usr/include \
  --with-db-lib=/usr/lib
  ```

This tells SquidGuard to set itself up for this particular configuration setup with:

- The location of where the squidGuard configuration file must be installed.
- The location of where the squidGuard log file must be installed.
- The location of where the squidGuard database directory must be installed.
- The location of the Berkley DB includes files that SquidGuard need.
- The location of the Berkley DB library that SquidGuard need.

Step 3
Now, we must make a list of all files on the system before installing the software, and one afterwards, then compare them using the **diff** utility to find out what files are placed where and finally we install the SquidGuard software:

```
[root@deep squidGuard-1.2.0]# make
[root@deep squidGuard-1.2.0]# cd
[root@deep root]# find /* > SquidGuard1
[root@deep root]# cd /var/tmp/squidGuard-1.2.0/
[root@deep squidGuard-1.2.0]# make install
[root@deep squidGuard-1.2.0]# cd samples/
[root@deep samples]# install -m 511 squidGuard.cgi /home/httpd/cgi-bin/
[root@deep samples]# cd dest/
[root@deep dest]# mkdir -p /var/spool/squid/squidGuard
[root@deep dest]# chown -R squid.squid /var/spool/squid/squidGuard/
[root@deep dest]# chmod 0750 /var/spool/squid/squidGuard/
[root@deep dest]# chown -R squid.squid /var/log/squid/squidGuard/
[root@deep dest]# chmod 0750 /var/log/squid/squidGuard/
[root@deep dest]# cp blacklists.tar.gz /var/spool/squid/squidGuard/
[root@deep dest]# cd /var/spool/squid/squidGuard/
[root@deep squidGuard]# mkdir -p aggressive
[root@deep squidGuard]# mkdir -p gambling
[root@deep squidGuard]# mkdir -p hacking
[root@deep squidGuard]# mkdir -p porn
[root@deep squidGuard]# chown -R squid.squid aggressive/
[root@deep squidGuard]# chown -R squid.squid gambling/
[root@deep squidGuard]# chown -R squid.squid hacking/
[root@deep squidGuard]# chown -R squid.squid porn/
[root@deep squidGuard]# tar xzpf blacklists.tar.gz
[root@deep squidGuard]# cd blacklists
[root@deep blacklists]# install -m 644 aggressive/domains
../aggressive/
[root@deep blacklists]# install -m 644 aggressive/urls ../aggressive/
[root@deep blacklists]# install -m 644 gambling/domains ../gambling/
[root@deep blacklists]# install -m 644 gambling/urls ../gambling/
[root@deep blacklists]# install -m 644 hacking/domains ../hacking/
[root@deep blacklists]# install -m 644 hacking/urls ../hacking/
[root@deep blacklists]# install -m 644 porn/domains ../porn/
[root@deep blacklists]# install -m 644 porn/urls ../porn/
[root@deep blacklists]# install -m 644 porn/expressions ../porn/
[root@deep blacklists]# cd ..
[root@deep squidGuard]# chown -R squid.squid *
[root@deep squidGuard]# strip /usr/bin/squidGuard
[root@deep squidGuard]# /sbin/ldconfig
[root@deep squidGuard]# rm -rf blacklists blacklists.tar.gz
[root@deep squidGuard]# cd
[root@deep root]# find /* > SquidGuard2
[root@deep root]# diff SquidGuard1 SquidGuard2 > SquidGuard-Installed
```

The **make** command will compile all source files into executable binaries that can be installed, and **make install** will install the binaries and any supporting files into the appropriate locations. The "**install -m 0511**" command will install the CGI program called squidGuard.cgi (squidGuard.cgi is a small script, which is used to explain to the user that the URL is blocked and by which rule set) into your cgi-bin directory.

The "**mkdir -p**" command will create the SquidGuard directory and subdirectories to store database filter files to run with squidGuard, the "**chown** and **chmod**" commands will set the appropriate mode and ownership permissions to the squidGuard directory and it's subdirectories. The "**tar**" command will untar the blacklists.tar.gz compressed archive containing all the filter files and the "**install -m 644**" commands will install the entire filter files to their appropriate directories.

Finally, the **strip** command will reduce the size of the specified binaries for optimum performance and the "**rm -rf**" commands will remove the blacklists directory and archive file that we no longer need on our system.

Step 4
Once the configuration, optimization, compilation, and installation of the SquidGuard software have been accomplished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete SquidGuard and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf squidGuard-version/
[root@deep tmp]# rm -f squidGuard-version.tar.gz
```

The rm command as used above will remove all the source files we have used to compile and install SquidGuard. It will also remove the SquidGuard compressed archive from the /var/tmp directory.

Configuring SquidGuard

After SquidGuard has been built and installed successfully on your system, your next step is to configure and customize the squidGuard.conf file to suit your needs.

The parameters entered into the squidGuard configuration file (squidGuard.conf) will decide how the ACL should be applied and on which users, hosts, IP addresses, times, dates, destination, source, etc.

- ✓ /etc/squid/squidGuard.conf (The SquidGuard Configuration File)
- ✓ /home/httpd/cgi-bin/squidGuard.cgi (The squidGuard.cgi File)

/etc/squid/squidGuard.conf: The SquidGuard Configuration File

The /etc/squid/squidGuard.conf file is the main and only configuration file for squidGuard. It is not difficult to configure or understand but we have to understand the rules order if we want to have a working SquidGuard configuration file.

The SquidGuard configuration file (squidGuard.conf) has a structure that must be followed during its configuration. Next we show you the recommended structure for the configuration file and the order in which declarations are supposed to appear. There are six different possible declarations where five are optional and one required.

1. Path declarations (i.e. logdir and dbhome) (optional)
2. Time space declarations (i.e. time zones) (optional)
3. Source group declarations (i.e. clients) (optional)
4. Destination group declarations (i.e. URLs) (optional)
5. Rewrite rule group declarations (optional)
6. Access control rule declaration (required)

The "**Path declarations (1)**" is used to define the location of the `SquidGuard` logfiles directory and to define the base for relative list filenames, also known as filter files. This declaration is optional but recommended for clarity.

```
dbhome /var/spool/squid/squidGuard
logdir /var/log/squid/squidGuard
```

In the above declaration we specify the database directory from where all list filenames/filter files and their possible subdirectories, which are used to handle source and/or destination group information, should be located (`dbhome /var/spool/squid/squidGuard`). In the second option, "`logdir /var/log/squid/squidGuard`", we specify the directory from where the `SquidGuard` log files are stored. With the "Path declaration" lines, we can ensure that `SquidGuard` will find the both directories when it runs.

The "**Time space declarations (2)**" is used to define time or date rules that can be used in our `ACL` to limit Internet access times based on this declaration. The "Time space declarations" is optional and should be defined only if you think that you'll need to restrict Internet access based on time. In most enterprises and universities, this feature is useful to control Internet access to working hours.

```
time workhours {
    weekly mtwhf 08:30 - 16:30
}
```

In the above declaration we define a range of days and hours that we will later use in our configuration to limit employee access to the Internet. This is based on the days and hours defined in the time space declaration above. Many different specifications and combinations exist. In our example, we limit connection to days of the week (`weekly mtwhf`) between 08:30 AM to 16:30 PM (`08:30 - 16:30`) for everyone who is a member of the "`workhours`" name. Individual IP address, or an IP addresses range can also be put into the "`workhours`" name.

We begin our definition with a reserved word called "`time`" that the software recognizes as the time declaration, we give this declaration a name of our choice, "`workhours`", we then add another reserved word called "`weekly`", which allows us to enter day parameters (`mtwhf`) for m=mon, t =tue, w=wed, h=thu, f=fri, and finally include the time constraint (`08:30 - 16:30`) for each day.

> **NOTE:** The numeric time formats are important. For example, if you want to define 8:30, you must use 08:30 and not 8:30 for HH:MM.

The “**Source group declarations (3)**” is used to define the source on which our rules and ACL will be applied. This declaration is again optional but used when we want to define a different source for our network.

```
src internal {
    ip     192.168.1.0/24
}
```

In the above declaration we define with an IP address and net prefix (192.168.1.0/24) what our source network is and where it comes from (here, they come from our internal network). We start our definition with a reserved word called "src" that the software recognizes as a source declaration, again we give this declaration a name of our choice “internal”, and we add another reserved word called “ip”, which allows us to specify the origin as an IP address. In our case the IP address is defined as an IP/net prefix.

NOTE: Source group declarations are matched in the order they are defined. If you have defined only one source group (as we do in our example), then there is no problem, but if you have more than one source group declaration, you must consider the order they are defined”.

The “**Destination group declarations (4)**” is used to define the destination on which the rules and ACL will be applied. This declaration is another optional declaration and is used to control what can be viewed on the Internet. It is in this declaration that we can associate with our ACL the filters file containing the IP addresses and/or domain names that must be blocked depending on their contents.

```
dest aggressive {
    domainlist     aggressive/domains
    urllist        aggressive/urls
}

dest gambling {
    domainlist     gambling/domains
    urllist        gambling/urls
}

dest hacking {
    domainlist     hacking/domains
    urllist        hacking/urls
}

dest porn {
    domainlist     porn/domains
    urllist        porn/urls
    expressionlist porn/expressions
}

dest warez {
    domainlist     warez/domains
    urllist        warez/urls
}
```

The above declarations are not difficult to understand. We can observe that we have five different destination groups defined. The specifications are the same only paths and filter names change.

Let's look at these in more detail. The reserved word called "dest" starts each of our groups and the software recognizes it as a destination declaration, we give this declaration a name of our choice, in this example it's called "agressive", and two other reserved words "domainlist" and "urllist".

The program interprets the "domainlist" specification as a path pointing to a file called "domains" located under the "/var/spool/squid/squidGuard/aggressive" directory, which contains all the domain names that must be blocked to users.

The program also interprets the "urllist" specification as the path pointing to a file called "urls" which is located under the "/var/spool/squid/squidGuard/aggressive" directory, which contains all the URL's that must be blocked and not accessible to the users.

In the above example, another specification exists, which is "expressionlist" that lets us specify via the "/var/spool/squid/squidGuard/porn/expressions" file, a list of regular expressions to use in the scan for blocked sites.

WARNING: As with the previous groups, declarations are matched in the order they are listed in the "pass" declaration. If you have defined only one destination group, then there is no problem, but if you have more than one destination group declaration, you must consider the order in which they will be listed during the configuration of your "Access control rule declaration". Regular expressions can produce bogus result in a search; it is up to you to decide if you really want to use regular expressions via the "expressionlist" file to block sites.

The "**Rewrite rule group declarations (5)**" is a special declaration option of SquidGuard that can be used to defined, for example via regular expressions, redirection to local copies within peek business hours of the most popular programs on the Internet. This declaration is optional and should be used with care since it can quite easily slow down SquidGuard on busy systems or produce bogus information. In our configuration, we don't use it.

The "**Access control rule declaration (6)**" is used to combine all of the previous declarations into distinct rulesets for each clientgroup. This is the place in our SquidGuard configuration, where our policies and ACL will take effect once properly defined.

```
acl {
     internal within workhours {
       pass !aggressive !gambling !hacking !porn !warez all
    }

    default {
       pass     none
       redirect http://gtw.openna.com/cgi-
bin/squidGuard.cgi?clientaddr=%a&clientname=%n&clientuser=%i&clientgroup=
%s&targetgroup=%t&url=%u
    }
}
```

In the above declaration, we inform the system what we want it to do when users try to connect to the Internet through the proxy server. This is our **A**ccess **C**ontrol **L**ists declaration. As with any of the previous declarations, we can see that the definition begins with a reserved word.

Therefore, we begin our definition with the reserved word called "acl" that the software recognizes as the beginning of our ACL definition. Next, we inform the program that this ACL applies to the source group called "internal", that we defined previously. We also inform it that this ACL applies within the company working hours we defined in the time space declaration section of the configuration.

We use the reserved word called "pass" to instruct it to allow users to view all Internet sites except ones in the blacklisted files "domains, urls, or expressions".

In other words, the "pass" rules declares destination groups that should pass for the actual client group. The "!" sign is the NOT operator and indicates a destination group that should not pass. It is good practice to always end the "pass" rule(s) with either "all" or "none" to make it/them clear.

We have another important section into our declaration called "default". The "default" rule set is used for all clients that match no clientgroup and for clientgroups with no acls declared. This section must always end our "acl" declaration for security reasons, since it will deny by default anything not previously declared and allowed. The "redirect" rule is used to redirect blocked destination groups, sites, users, etc to an alternative URL, where they will get more information about the reason why they cannot access the requested Internet site.

WARNING: You cannot define or use more than one acl block in the squidGuard.conf file. Everything must be defined in the same acl block.

Step1
Now that we have a better idea about how the SquidGuard configuration file works, it's time to think about what we need to define inside it. Let's create the SquidGuard configuration file.

Our example assumes that you want to permit Internet access during working hours for all internal client workstations coming from the IP address range 192.168.1.0/24, and that you want to deny access to aggressive, gambling, hacking, and porn sites and redirect any refused connections to an alternative URL.

This configuration is suitable for most needs. If you have a specific requirement, then you have to adjust the configuration and read the SquidGuard documentation for more information. For optimum security, we will merge the SquidGuard ACL with the Squid ACL to force clients to enter a username and password before accessing the Internet.

- Create the **squidGuard.conf** file (touch /etc/squid/squidGuard.conf) and add the following ACL lines:

```
dbhome /var/spool/squid/squidGuard
logdir /var/log/squid/squidGuard

# TIME SPACE DECLARATIONS
# The following declaration define a time rule from where clients are
# allowed and can access the Internet. Outside this time, connections
# will be denied.
#
time workhours {
    weekly mtwhf 08:30 - 17:30
}
```

```
# SOURCE GROUP DECLARATIONS
# The following declaration define a source group, or client groups IP
# addresses range from where connection to the Internet through the proxy
# are allowed.
#
src internal {
    ip     192.168.1.0/24
}


# DESTINATION GROUP DECLARATIONS
# The following declaration define destination group, or target groups
# websites where connection are forbiden.
#
dest aggressive {
    domainlist     aggressive/domains
    urllist        aggressive/urls
}

dest gambling {
    domainlist     gambling/domains
    urllist        gambling/urls
}

dest hacking {
    domainlist     hacking/domains
    urllist        hacking/urls
}

dest porn {
    domainlist     porn/domains
    urllist        porn/urls
    expressionlist porn/expressions
}


# REWRITE RULES GROUP DECLARATIONS
#


# ACCESS CONTROL LISTS
# The Access Control List, ACL, combines the previous definitions into
# distinct rulesets for each clientgroup.
#
acl {
     internal within workhours {
       pass !aggressive !gambling !hacking !porn all
    }

    default {
       pass     none
       redirect http://my.proxy.com/cgi-
bin/squidGuard.cgi?clientaddr=%a&clientname=%n&clientuser=%i&clientgroup=
%s&targetgroup=%t&url=%u
    }
}
```

Step2
Once the SquidGuard has been configured, we have to include in our default Squid configuration file some additional lines that will make Squid Proxy Server run with SquidGuard. In the configuration below, we use the default squid.conf file as described in the Squid chapter. The text in bold are the parts of the configuration file that we have added to the default Squid configuration file as used in the Squid chapter.

- Edit the **squid.conf** file (vi /etc/squid/squid.conf) and add the following options to make Squid runs with SquidGuard:

```
icp_port 0
hierarchy_stoplist cgi-bin ?
acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY
cache_mem 128 MB
cache_replacement_policy heap GDSF
memory_replacement_policy heap GDSF
cache_dir diskd /var/spool/squid 2000 16 256
cache_store_log none
log_fqdn on
redirect_program /usr/bin/squidGuard
acl localnet src 192.168.1.0/255.255.255.0
acl localhost src 127.0.0.1/255.255.255.255
acl Safe_ports port 80 443 210 70 21 1025-65535
acl CONNECT method CONNECT
acl all src 0.0.0.0/0.0.0.0
http_access allow localnet
http_access allow localhost
http_access deny !Safe_ports
http_access deny CONNECT
http_access deny all
cache_mgr sysadmin@openna.com
cache_effective_user squid
cache_effective_group squid
logfile_rotate 0
log_icp_queries off
cachemgr_passwd my-secret-pass all
buffered_logs on
```

The option "redirect_program", specifies the location of the URL redirector executable. The executable in our case is the squidguard binary program located under the "/usr/bin" directory. Once the "redirect_program" line is added into the squid.conf file, Squid will know that it must run and work with a new program called squidguard. In this way, Squid will continue its proxy job and SuidGuard will be in charge filtering, checking, authorizing and redirecting, if necessary, all Internet destinations.

The option "log_fqdn", enables reverse lookups with Squid. This is important with SquidGuard, since the use of domain matches for clientsgroups requires that Squid is set up to do reverse lookups on clients. Without this option, any domain specification parameters in the SquidGuard configuration file that point to a filter file will simply not work. Therefore, when SquidGuard is used with Squid, we have to check and enable this option in the squid.conf file.

Step3
For additional security or for those who want to authenticate users with a username and password before allowing Internet access, there are some previously shown options that we can add into our `squid.conf` file. Below, we use the `squid.conf` file used in step 2 and add the user authentication feature. The text in bold are the parts of the configuration file that we have added to the above `Squid` configuration file.

- Edit the **`squid.conf`** file (`vi /etc/squid/squid.conf`) and add the following options to make `Squid` use the users authentication feature:

```
icp_port 0
hierarchy_stoplist cgi-bin ?
acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY
cache_mem 128 MB
cache_replacement_policy heap GDSF
memory_replacement_policy heap GDSF
cache_dir diskd /var/spool/squid 2000 16 256
cache_store_log none
log_fqdn on
redirect_program /usr/bin/squidGuard
authenticate_program /usr/lib/squid/pam_auth /etc/passwd
acl authenticated proxy_auth REQUIRED
acl localnet src 192.168.1.0/255.255.255.0
acl localhost src 127.0.0.1/255.255.255.255
acl Safe_ports port 80 443 210 70 21 1025-65535
acl CONNECT method CONNECT
acl all src 0.0.0.0/0.0.0.0
http_access allow authenticated
http_access allow localnet
http_access allow localhost
http_access deny !Safe_ports
http_access deny CONNECT
http_access deny all
cache_mgr sysadmin@openna.com
cache_effective_user squid
cache_effective_group squid
logfile_rotate 0
log_icp_queries off
cachemgr_passwd my-secret-pass all
buffered_logs on
```

NOTE: If you need more information about Users Authentication support with `Squid`, please see the previous `Squid` chapter.

/home/httpd/cgi-bin/squidGuard.cgi: The SquidGuard.cgi File

The squidGuard.cgi program is what users will see if the sites they try to access are blocked by company policy. It is used to explain to the user that the URL is blocked and by which rule set.

Step1
There are two important options to configure in this small cgi program to make it work for your site. Below we show you how to do it.

- Edit the **squidGuard.cgi** program (vi /home/httpd/cgi-bin/squidGuard.cgi) and change the following options to make SquidGuard runs for your site:

```
$proxy       =        "my.proxydomain.com";
$proxymaster =    "sysadmin\@proxydomain.com";
```

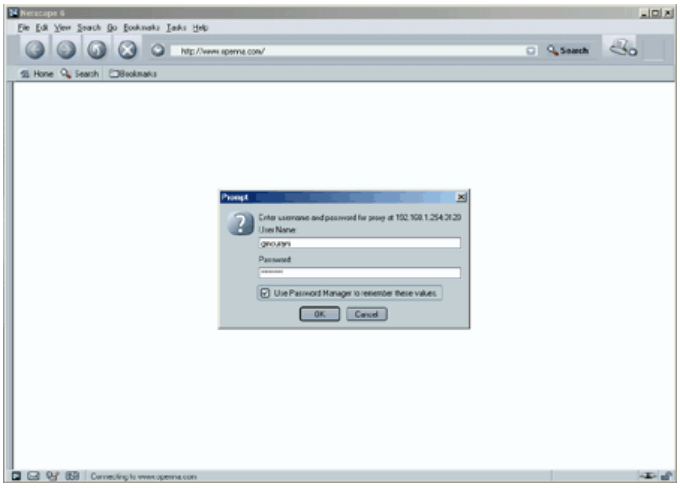
Where "my.proxydomain.com" is the FQDN of the Gateway Server where SquidGuard is running, and "sysadmin\@proxydomain.com" is the email address of the administrator.

> **NOTE:** You can use any personal html page of your choice to replace the squidGuard.cgi script, if it does not fit with your requirements. There is no problem as long as your squidGuard configuration file is properly updated to point to the right file.

Testing SquidGuard

Now it is time to restart our Squid server for all the changes to take effect and connect to the Internet with our preferred browser to see if everything is working as expected.

1. First, we try to connect to a legitimate site. We should receive a new window asking us to enter username and password.



2. Now we will try to access a blocked warez site just to see if SquidGuard filtering works as expected. For example, try www.warez.com. We must be redirected to a new URL, which will give us the reason why we cannot access this site.



NOTE: If you receive an error message here, it is likely because you have forgot to configure the `squidguard.cgi` program to fit your domain name information. Please edit the "`/home/httpd/cgi-bin/suidguard.cgi`" program and make the appropriate changes.

Optimizing `SquidGuard`

This section deals specifically with the actions we can take to improve and tighten the performance of `SquidGuard`. Note that we refer to the features available within the base installed program only.

Creating a prebuilt database

Ok, your `SquidGuard` program is running and you're happy to see that it works, but wait a minute; it is possible to make `SquidGuard` runs faster simply by converting its filter files (where blacklisted domains and urls reside) into a `db` filter file.

The default filter files used with `SquidGuard` are in plain text format, and `SquidGuard` needs to parse all the lines inside the filter file to decide if domains/url's can be allowed or not. There is a better method that gives the same result and also runs faster by converting all of its filter files into a `db` file.

Step1
The first step to accomplish this conversion will be to use the "`-C`" command of `SquidGuard`. This command simply converts the text file into a `db` file.

- To convert your filter text files into a db file,use the following commands:

```
[root@deep /]# cd /var/spool/squid/squidGuard/
[root@deep /]# squidGuard -C aggressive/domains
[root@deep /]# squidGuard -C aggressive/urls
[root@deep /]# squidGuard -C gambling/domains
[root@deep /]# squidGuard -C gambling/urls
[root@deep /]# squidGuard -C hacking/domains
[root@deep /]# squidGuard -C hacking/urls
[root@deep /]# squidGuard -C porn/domains
[root@deep /]# squidGuard -C porn/urls
```

The above commands, will convert a domainlist or urllist from plain text file to a prebuilt database.

NOTE: There is one filter file that cannot and should not be converted into a `db` file. This filter file is the "`expressions`" file located under the "`porn`" directory.

Step2
Once all of our filter files have been converted, now we have to edit our `squidGuard.conf` file to change the default extension for our filter files to reflect the change. The text in bold are the parts of the configuration file that we have changed in the default `SquidGuard` configuration file.

- Edit the **`squidGuard.conf`** file (`vi /etc/squid/squidGuard.conf`) and change the following lines to make `SquidGuard` load and interpret the entire db file now.

```
dbhome /var/spool/squid/squidGuard
logdir /var/log/squid/squidGuard

# TIME SPACE DECLARATIONS
# The following declaration define a time rule from where clients are
# allowed and can access the Internet. Outside this time, connections
# will be denied.
#
time workhours {
    weekly mtwhf 08:30 - 17:30
}


# SOURCE GROUP DECLARATIONS
# The following declaration define a source group, or client groups IP
# addresses range from where connection to the Internet through the proxy
# are allowed.
#
src internal {
    ip     192.168.1.0/24
}


# DESTINATION GROUP DECLARATIONS
# The following declaration define destination group, or target groups
# websites where connection are forbiden.
#
dest aggressive {
    domainlist     aggressive/domains.db
    urllist        aggressive/urls.db
}

dest gambling {
    domainlist     gambling/domains.db
    urllist        gambling/urls.db
}

dest hacking {
    domainlist     hacking/domains.db
    urllist        hacking/urls.db
}

dest porn {
    domainlist     porn/domains.db
    urllist        porn/urls.db
    expressionlist porn/expressions
}
```

```
# REWRITE RULES GROUP DECLARATIONS
#


# ACCESS CONTROL LISTS
# The Access Control List, ACL, combines the previous definitions into
# distinct rulesets for each clientgroup.
#
acl {
     internal within workhours {
       pass !aggressive !gambling !hacking !porn all
    }

    default {
       pass     none
       redirect http://my.proxy.com/cgi-
bin/squidGuard.cgi?clientaddr=%a&clientname=%n&clientuser=%i&clientgroup=
%s&targetgroup=%t&url=%u
    }
}
```

Step3
Finally, we have to restart our `Squid` Proxy server for the changes to take effect.

CHAPTER 13

FreeS/WAN VPN

IN THIS CHAPTER

1. Compiling - Optimizing & Installing `FreeS/WAN`
2. Configuring `FreeS/WAN`
3. Configuring `RSA` private keys secrets
4. Requiring network setup for `IPSec`
5. Testing the `FreeS/WAN` installation

Linux FreeS/WAN

Abstract

First of, I would like to mention that this chapter about FreeSWAN is an unsupported chapter now. This because FreeSWAN is a very special piece of software that often required specific kernel versions to work on the system. Since kernel versions are updated frequently and more often than FreeSWAN versions, there is no guarantee that the kernel version you use when reading this chapter will be compatible with FreeSWAN. Also, FreeSWAN is not software that everyone uses daily on the Internet for proper operation of their servers.

Usually, only experts and companies, which have specific needs for their network, will need to install and use it. For this reason, I've decided to not provide advanced information about FreeSWAN in this book but since some of you will certainly ask for it, I'll provide some information about how to compile, configure and run it for Linux. Unlike other chapters in this book, there is no guarantee that the information provided here will work for your system. If you have problem getting FreeSWAN to work for you, then ask the FreeSWAN group for some help. Here is just some basic startup information about FreeSWAN now.

Protection of client-to-server and vice versa with PGP for mail, SSH for remote login, and SSL solutions are an excellent choice but sometimes for enterprise environments establishing secure communication channels, assuring full privacy, authenticity and data integrity between two gateway machines, routers, or firewalls system over the Internet are vital. For this, IPSEC has been created.

IPSEC is **I**nternet **P**rotocol **SEC**urity. It uses strong cryptography to provide both authentication and encryption services. Authentication ensures that packets are from the right sender and have not been altered in transit. Encryption prevents unauthorized reading of packet contents. IPSEC can protect any protocol running above IP and any medium used below IP.

IPSEC can also provide some security services "in the background", with no visible impact on users. More to the point, it can protect a mixture of protocols running over a complex combination of media (i.e. IMAP/POP etc.) without having to change them in any way, since the encryption occurs at the IP level.

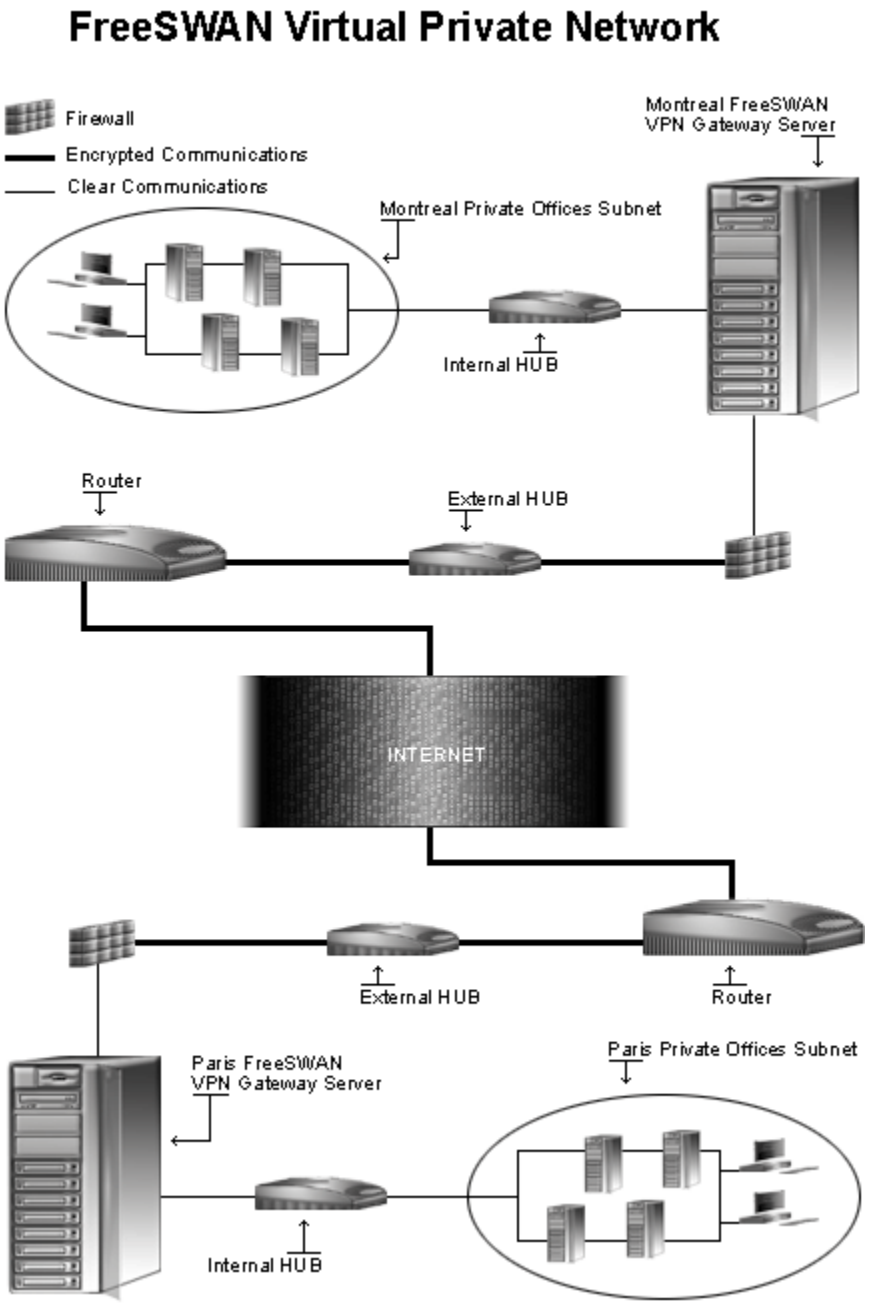
Three protocols are used with FreeS/WAN to archive this result:

1. AH (**A**uthentication **H**eader) provides a packet-level authentication service.
2. ESP (**E**ncapsulating **S**ecurity **P**ayload) provides encryption plus authentication.
3. IKE (**I**nternet **K**ey **E**xchange) negotiates connection parameters.

The FreeSWAN implementation has three main parts:

1. KLIPS (**k**erne**l** **IPs**ec) implements AH, ESP, and packet handling within the kernel.
2. Pluto (an IKE daemon) implements IKE, negotiating connections with other systems.
3. Various scripts provide an adminstrator's interface to the machinery.

IPSEC services allow you to build secure tunnels through untrusted networks like the Internet. Everything passing through the untrusted net is encrypted by the IPSEC gateway machine and decrypted by the gateway server at the other end. The result is **V**irtual **P**rivate **N**etwork or VPN. This is a network, which is effectively private even though it includes machines at several different sites connected by the insecure Internet.


FreeSWAN Virtual Private Network
Firewall
Encrypted Communications
Clear Communications
Montreal FreeSWAN
VPN Gateway Server
Montreal Private Offices Subnet
Internal HUB
Router
External HUB
INTERNET
External HUB
Router
Paris FreeSWAN
VPN Gateway Server
Paris Private Offices Subnet
Internal HUB

These installation instructions assume

Commands are Unix-compatible.
The source path is /usr/src (note that other paths are possible, at personal discretion).
Installations were tested on OpenNA Linux and Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: Yes
Latest FreeS/WAN version number is 1.95

The procedures given in this chapter are likely to work on all Linux platforms, but we have only tested it on OpenNA Linux and Red Hat Linux.

Packages

The following is based on information listed by FreeS/WAN as of 2001/02/04. Please check http://www.freeswan.org/ regularly for the latest status. We chose to install from source because it provides the facility to fine tune the installation.

Source code is available from:
FreeS/WAN Homepage Site: http://www.freeswan.org/
FreeS/WAN FTP Site: 194.109.6.26
You must be sure to download: freeswan-1.95.tar.gz

Prerequisites

Linux FreeS/WAN requires that the software below is already installed on your system to be able to run and work successfully. If this is not the case, you must install it from your Linux CD-ROM or source archive file. Please make sure you have this program installed on your machine before you proceed with this chapter.

- ✓ gmp is required to compile and make FreeS/WAN works in your system.

NOTE: Not installing the GMP library will make pluto fail to compile on your server.

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all the files installed onto the system if you want to update the package in the future. To solve this problem, it's a good idea to make a list of files on the system before you install FreeS/WAN, and then one afterwards, and then compares them using the **diff** utility to find out what files were placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep root]# find /* > Freeswan1
  ```
- And the following one after you install the software:
  ```
  [root@deep root]# find /* > Freeswan2
  ```
- Then use this command to get a list of what changed:
  ```
  [root@deep root]# diff Freeswan1 Freeswan2 > Freeswan-Installed
  ```

Compiling - Optimizing & Installing FreeS/WAN

Below are the required steps that you must make to compile and optimize the FreeS/WAN software before installing it into your Linux system.

Step 1
The installation of IPSEC FreeS/WAN software requires some modification of your original kernel since some parts of FreeS/WAN must be included and incorporated in your kernel before you can use it. For this reason the first step in installing FreeS/WAN is to go to the Linux Kernel section in this book and follow the instructions on how to install the Linux Kernel on your system (even if you have already done this before) and come back to this section after you have executed the "**make dep; make clean**" commands, but before the "**make bzImage**" command in the Linux Kernel chapter.

Step 2
Once your kernel is configured and you download the FreeS/WAN program from the main software site you must copy it to the /usr/src directory and change to this location before expanding the archive. Putting FreeS/WAN under /usr/src/linux will confuse the links, therefore, expand the software under /usr/src and never under /usr/src/linux directory.

- These procedures can be accomplished with the following commands:

```
[root@deep /]# cp freeswan-version.tar.gz /usr/src/
[root@deep /]# cd /usr/src/
[root@deep src]# tar xzpf freeswan-version.tar.gz
```

Step 3
After that, move into the newly created FreeS/WAN directory then configure, compile and optimize it.

- To move into the top-level directory of FreeS/WAN distribution use the command:

```
[root@deep src]# cd freeswan-1.95/
```

Step 4
You must modify the Makefile.inc under the FreeS/WAN source directory to specify installation paths and optimization parameters. We must modify this file to be compliant with Linux file system structure, add optimization parameters for our specific processor architecture and install FreeS/WAN files under our PATH environment variable.

- Edit the **Makefile.inc** file (vi Makefile.inc) and change all of the targeted lines in the order shown below:

```
PUBDIR=$(DESTDIR)/usr/local/sbin
```

To read:

INC_USRLOCAL=/usr

```
REALPRIVDIR=/usr/local/lib/ipsec
```

To read:

INC_MANDIR=/share/man

```
MANTREE=$(DESTDIR)/usr/local/man
```

To read:

```
USERCOMPILE=-O2
```

```
CONFDIR=$(DESTDIR)/etc
```

To read:

```
KLIPSCOMPILE=-O2
```

All of the above changes, will relocate all files related to the FreeS/WAN software to the destination target directories we have chosen. We also add optimization parameters related to the type of processor that we use in our system for better performance.

Step 5
Once the modifications have been made to the source file of FreeS/WAN as described in step 4, we need to patch the pre-configured Linux Kernel to include FreeS/WAN support.

- This procedure can be accomplished with the following command:

```
[root@deep freeswan-1.95]# make ogo
echo "===============" >>out.kpatch
echo "`date` `cd /usr/src/linux ; pwd`" >>out.kpatch
make _patches2.3 >>out.kpatch
……
```

The **make ogo** command is what we use to patch the kernel. It will automatically start the kernel configuration part for the second time and will let you answer all kernel configuration questions before compilation and integration of its component into the kernel.

During the second kernel configuration, be sure that your kernel has been built with FreeS/WAN support enabled. A new section related to Frees/WAN support named "IPSec options (FreeS/WAN)" should appear in your kernel configuration after you have patched the kernel with the FreeS/WAN program as described above. You need ensure that you have answered **Y** to the following questions under the new kernel section: IPSec options (FreeS/WAN).

```
IP Security Protocol (FreeS/WAN IPSEC) (CONFIG_IPSEC) [Y/n/?]
*
* IPSec options (FreeS/WAN)
*
   IPSEC: IP-in-IP encapsulation (tunnel mode) (CONFIG_IPSEC_IPIP) [Y/n/?]
   IPSEC: Authentication Header (CONFIG_IPSEC_AH) [Y/n/?]
   HMAC-MD5 authentication algorithm (CONFIG_IPSEC_AUTH_HMAC_MD5) [Y/n/?]
   HMAC-SHA1 authentication algorithm (CONFIG_IPSEC_AUTH_HMAC_SHA1) [Y/n/?]
   IPSEC: Encapsulating Security Payload (CONFIG_IPSEC_ESP) [Y/n/?]
   3DES encryption algorithm (CONFIG_IPSEC_ENC_3DES) [Y/n/?]
   IPSEC: IP Compression (CONFIG_IPSEC_IPCOMP) [Y/n/?]
   IPSEC Debugging Option (CONFIG_IPSEC_DEBUG) [Y/n/?]
```

NOTE: All the customization you made to your kernel the first time you ran the `make config`, `make dep`, and `make clean` commands will be preserved, so you don't need to reconfigure every part of your kernel; Just the new section added by `FreeS/WAN` named "`IPSec options (FreeS/WAN)`" is required, as shown above.

Some networking options will get turned on automatically, even if you previously turned them off; This is because `IPSEC` needs them. Whichever configuration program you are using, you should pay careful attention to a few issues: in particular, do NOT disable any of the following under the "`Networking Options`" of your kernel configuration:

```
Kernel/User netlink socket (CONFIG_NETLINK) [Y/n/?]
Netlink device emulation (CONFIG_NETLINK_DEV) [Y/n/?]
```

Step 6
Once the **`make ogo`** command is completed, your `FreeS/WAN` software and Linux kernel with `FreeS/WAN` support is ready to be installed on your server. We must make a list of files on the system before you install the software and the kernel, and one afterwards, then compare them using the **`diff`** utility to find out what files are placed where and finally install `FreeS/WAN` and the new kernel with `FreeS/WAN` support in your server:

```
[root@deep freeswan-1.95]# cd
[root@deep root]# find /* > Freeswan1
[root@deep root]# cd /usr/src/freeswan-1.95/
[root@deep freeswan-1.95]# make install
[root@deep freeswan-1.95]# cd
[root@deep root]# find /* > Freeswan2
[root@deep root]# diff Freeswan1 Freeswan2 > Freeswan-Installed
```

The **`make install`** command will install all `FreeS/WAN` and kernel components together to the appropriated location on your server.

Step 7
At this stage of your installation of `FreeS/WAN`, you must follow the rest of the instructions in the Linux Kernel chapter of this book as normal to install the kernel. At this point, after you have copied and installed your new kernel image, `system.map`, or modules (if necessary), and set the `lilo.conf` file to load the new kernel, you must edit and customize the configuration files related to `FreeS/WAN` "`ipsec.conf`" and "`ipsec.secrets`" before rebooting your system.

Step 8
Once the compilation, optimization and installation of the software have been finished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete `FreeS/WAN` and its related source directory, use the following commands:

```
[root@deep /]# cd /usr/src/
[root@deep src]# rm -rf freeswan-version/
[root@deep src]# rm -f freeswan-version.tar.gz
```

Configuring FreeS/WAN

After building FreeS/WAN, your next step is to verify or change, if necessary, the options in your FreeS/WAN configuration files. Those files are:

- ✓ /etc/ipsec.conf (The FreeS/WAN Configuration File)
- ✓ /etc/ipsec.secrets (The FreeS/WAN Configuration File to store secret keys)

/etc/ipsec.conf: The FreeS/WAN Configuration File

The configuration file for FreeS/WAN (/etc/ipsec.conf) allows you to configure your IPSEC configurations, control information and connections types. IPSEC currently supports two types of connections: Manually keyed and Automatically keyed.

The difference is strictly in how they are keyed. Manually keyed connections use keys stored in the /etc/ipsec.conf file. This type of connection is less secure than automatically keyed. Automatically keyed connections use keys automatically generated by the Pluto key negotiation daemon. The key negotiation protocol, used by default and named IKE, authenticates the other system using shared secrets stored in /etc/ipsec.secrets file. For these reasons, we will use the automatically keyed connection that is more secure than the manually keyed connection (it is highly recommended that you use the automatically keyed connection).

In our example configuration below, we configure a sample tunnel with a firewall-penetrating tunnel, and we assume that firewalling is being done on the left and right side. We choose to show you this configuration since we assume it is what most users and companies will use.

Also, it allows us to play with more options in the configuration file ipsec.conf for automatically keyed connections. Different configurations exist and you may consult the "doc/examples" file under the subdirectory "doc" of the Frees/WAN source directory for more information and other possible configurations.

We must edit the **ipsec.conf** file (vi /etc/ipsec.conf) and change the default values to fit our specifications for IPSEC configuration and communication. Currently there are two types of section in this file (/etc/ipsec.conf): a "**config**" section, which specifies general configuration information for IPSEC, and a "**conn**" section which specifies an IPSEC connection. Its contents are not security-sensitive unless manual keying is being done (recall, manual keying is not recommended for security reasons).

The first section type, named **config setup,** is the only **config** section known to the IPSEC software containing overall setup parameters for IPSEC that applies to all connections, and information used when the software is being started.

The second type, named **conn,** contains a connection specification defining a network connection to be made using IPSEC. The name it is given is arbitrary, and is simply used to identify the connection to ipsec_auto(8) and ipsec_manual(8).

```
# /etc/ipsec.conf - FreeS/WAN IPSEC configuration file

# More elaborate and more varied sample configurations can be found
# in doc/examples.

# basic configuration
config setup
        interfaces="ipsec0=eth0"
        klipsdebug=none
        plutodebug=none
        plutoload=%search
        plutostart=%search

# sample connection
conn deep-mail
        left=208.164.186.1
        leftsubnet=192.168.1.0/24
        leftnexthop=205.151.222.250
        right=208.164.186.2
        rightsubnet=192.168.1.0/24
        rightnexthop=205.151.222.251
        keyingtries=0
        auth=ah
        auto=start
```

This tells the `ipsec.conf` file to set itself up for this particular configuration setup with:

`interfaces="ipsec0=eth0"`
This option specifies which appropriate virtual and physical interfaces for `IPSEC` to use. The default setting, "`interfaces=%defaultroute`", will look for your default connection to the Internet, or your corporate network. Also, you can name one or more specific interfaces to be used by `FreeS/WAN`. For example:

```
interfaces="ipsec0=eth0"
interfaces="ipsec0=eth0 ipsec1=ppp0"
```

Both set the `eth0` interface as `ipsec0`. The second one, however, also supports `IPSEC` over a `PPP` interface. If the default setting "`interfaces=%defaultroute`" is not used, then the specified interfaces will be the only ones this gateway machine can use to communicate with other `IPSEC` gateways.

`klipsdebug=none`
This option specifies the debugging output for `KLIPS` (the kernel `IPSEC` code). The default value **`none`**, means no debugging output and the value **`all`** means full output.

`plutodebug=none`
This option specifies the debugging output for the `Pluto` key. The default value, **`none`**, means no debugging output, and the value **`all`** means full output.

`plutoload=%search`
This option specifies which connections (by name) to load automatically into memory when `Pluto` starts. The default is none and the value **`%search`** loads all connections with `auto=add` or `auto=start`.

`plutostart=%search`
This option specifies which connections (by name) to automatically negotiate when `Pluto` starts. The default is none and the value **`%search`** starts all connections with `auto=start`.

conn deep-mail
This option specifies the name given to identify the connection specification to be made using IPSEC. It's a good convention to name connections by their ends to avoid mistakes. For example, the link between deep.openna.com and mail.openna.com gateways server can be named "deep-mail", or the link between your Montreal and Paris offices, "montreal-paris".

Note that the name "deep-mail" or whatever you have chosen should be the same in the ipsec.conf file on both gateways. In other words, the only change you should make in the /etc/ipsec.conf file on the second gateway is changing the "interfaces=" line to match the interface the second gateway uses for IPSEC connection, if, of course, it's different from the first gateway. For example, if the interface eth0 is used on the both gateways for IPSEC communication, you don't need to change the line "interfaces=" on the second gateway. On the other hand, if the first gateway uses eth0 and the second uses eth1, you must change the line "interfaces=" on the second gateway to match the interface eth1.

left=208.164.186.1
This option specifies the IP address of the gateway's external interface used to talk to the other gateway.

leftsubnet=192.168.1.0/24
This option specifies the IP network or address of the private subnet behind the gateway.

leftnexthop=205.151.222.250
This option specifies the IP address of the first router in the appropriate direction or ISP router.

right=208.164.186.2
This is the same explanation as "left=" but for the right destination.

rightsubnet=192.168.1.0/24
This is the same explanation as "leftsubnet=" but for the right destination.

rightnexthop=205.151.222.251
This is the same explanation as "leftnexthop=" but for the right destination.

keyingtries=0
This option specifies how many attempts (an integer) should be made in (re)keying negotiations. The default value 0 (retry forever) is recommended.

auth=ah
This option specifies whether authentication should be done separately using AH (Authentication Header), or be included as part of the ESP (Encapsulated Security Payload) service. This is preferable when the IP headers are exposed to prevent man-in-the-middle attacks.

auto=start
This option specifies whether automatic startup operations should be done at IPSEC startup.

NOTE: A data mismatch anywhere in this configuration "ipsec.conf" will cause FreeS/WAN to fail and to log various error messages.

/etc/ipsec.secrets: The FreeS/WAN File to store Secret Keys

The file ipsec.secrets stores the secrets used by the pluto daemon to authenticate communication between both gateways. Two different kinds of secrets can be configured in this file, preshared secrets and RSA private keys. You must check the permissions of this file to be sure that the super-user “root” owns the file, and its permissions are set to block all access by others.

Step 1
An example secret is supplied in the ipsec.secrets file by default. You should change it by creating your own. With automatic keying you may have a shared secret up to 256 bits, which is then used during the key exchanges to make sure a man in the middle attack does not occur.

- To create a new shared secret, use the following commands:
 [root@deep /]# **ipsec ranbits 256 > temp**

New, random keys are created with the ranbits(8) utility in the file named “temp”. The ranbits utility may pause for a few seconds if not enough entropy is available immediately. Don’t forget to delete the temporary file as soon as you are done with it.

Step 2
Now that our new shared secret key has been created in the “temp” file, we must put it in the /etc/ipsec.secrets file. When editing the ipsec.secrets file, you should see something like the following appearing in your text editor. Each line has the IP addresses of the two gateways plus the secret. It should look something like this:

```
# This file holds shared secrets which are currently the only inter-Pluto
# authentication mechanism.  See ipsec_pluto(8) manpage.  Each secret is
# (oversimplifying slightly) for one pair of negotiating hosts.

# The shared secrets are arbitrary character strings and should be both
# long and hard to guess.

# Note that all secrets must now be enclosed in quotes, even if they have
# no white space inside them.

10.0.0.1 11.0.0.1         "jxVS1kVUTTulkVRRTnTujSm444jRuU1mlkklku2nkW3nnVu
V2WjjRRnulmlkmU1Run5VSnnRT"
```

- Edit the **ipsec.secrets** file (vi /etc/ipsec.secrets) and change the default secrets keys:

```
10.0.0.1 11.0.0.1  " jxVS1kVUTTulkVRRTnTujSm444jRuU1mlkklku2nkW3nnVu
V2WjjRRnulmlkmU1Run5VSnnRT "
```

To read:

```
208.164.186.1 208.164.186.2
"0x9748cc31_2e99194f_d230589b_cd846b57_dc070b01_74b66f34_19c40a1a_804906ed"
```

Where “208.164.186.1" and “208.164.186.2" are the IP addresses of the two gateways and "0x9748cc31_2e99194f_d230589b_cd846b57_dc070b01_74b66f34_19c40a1a_804906ed" (note that the quotes are required) is the shared secret we have generated above with the command “ipsec ranbits 256 > temp” in the “temp” file.

Step 3
The files ipsec.conf, and ipsec.secrets must be copied to the second gateway machine so as to be identical on both ends. The only exception to this is the ipsec.conf file, which must have in it a section labeled by the line **config setup** with the correct interface settings for the second gateway, if they differ from the first. The ipsec.secrets file, contrary to the RSA private key, should have the same-shared secrets on the two gateways.

WARNING: The file /etc/ipsec.secrets should have permissions rw------- (600) and be owned by the super-user "root". The file /etc/ipsec.conf is installed with permissions rw-r--r— (644) and must be owned also by "root".

Configuring RSA private keys secrets

Recall that currently with FreeSWAN software there are two kinds of secrets: preshared secrets and RSA private keys. The preshared secrets are what we have configured in our ipsec.conf and ipsec.secrets example, above. Some people may prefer to use RSA private keys for authentication by the Pluto daemon of the other hosts. If you are in this situation, you will have to make some minor modifications to your ipsec.conf and ipsec.secrets files as described in the following steps:

You need to create a separate RSA key for *each* gateway. Each one gets its private key in its own ipsec.secrets file, and the public keys go in leftrsasigkey and rightrsasigkey parameters in the **conn** description of ipsec.conf file, which goes to both.

Step 1
Create a separate RSA key for *each* gateway:

- On the first gateway (e.i. deep), use the following commands:

```
[root@deep /]# cd /
[root@deep /]# ipsec rsasigkey --verbose 1024 > deep-keys
computing primes and modulus...
getting 64 random bytes from /dev/random
looking for a prime starting there
found it after 30 tries
getting 64 random bytes from /dev/random
looking for a prime starting there
found it after 230 tries
swapping primes so p is the larger
computing (p-1)*(q-1)...
computing d...
computing exp1, exp1, coeff...
output...
```

- On the second gateway (e.i. mail), use the following commands:

```
[root@mail /]# cd /
[root@mail /]# ipsec rsasigkey --verbose 1024 > mail-keys
computing primes and modulus...
getting 64 random bytes from /dev/random
looking for a prime starting there
found it after 30 tries
getting 64 random bytes from /dev/random
looking for a prime starting there
found it after 230 tries
swapping primes so p is the larger
computing (p-1)*(q-1)...
```

```
computing d...
computing exp1, exp1, coeff...
output...
```

The `rsasigkey` utility generates an RSA public and private key pair of a 1024-bit signature, and puts it in the file `deep-keys` (`mail-keys` for the second command on the second gateway). The private key can be inserted verbatim into the `ipsec.secrets` file, and the public key into the `ipsec.conf` file.

WARNING: The `rsasigkey` utility may pause for a few seconds if not enough entropy is available immediately. You may want to give it some bogus activity such as random mouse movements. The temporary RSA "`deep-keys`" and "`mail-keys`" files should be deleted as soon as you are done with it. Don't forget to delete the `deep-keys` and `mail-keys` RSA files.

Step 2
Modify your `/etc/ipsec.conf` files to use RSA public keys in *each* gateway:

Edit you original **`ipsec.conf`** file (`vi /etc/ipsec.conf`) and add the following parameters related to RSA in the **`conn`** desciption of your `ipsec.conf` file on both gateway:

```
# sample connection
conn deep-mail
        left=208.164.186.1
        leftsubnet=192.168.1.0/24
        leftnexthop=205.151.222.250
        right=208.164.186.2
        rightsubnet=192.168.1.0/24
        rightnexthop=205.151.222.251
        keyingtries=0
        auth=ah
        authby=rsasig
        leftrsasigkey=<Public key of deep>
        rightrsasigkey=<Public key of mail>
        auto=start
```

`authby=rsasig`
This parameter specifies how the two security gateways should authenticate each other. The default value is secret for shared secrets. We must specify rsasig for RSA since we have decided to use RSA digital signatures.

`leftrsasigkey=<Public key of deep>`
This parameter specifies the left participant's public key for RSA signature authentication. In our example, left is `208.164.186.1`, and represents `deep.openna.com`, so we must put the RSA public key for `deep` on this line.

`rightrsasigkey=<Public key of mail>`
This parameter specifies the right participant's public key for RSA signature authentication. In our example, right is `208.164.186.2`, and represents `mail.openna.com`, so we must put the RSA public key of `mail` on this line.

You can retrieve the public key of `deep` in the RSA key file named "`deep-keys`", and the public key of `mail` in the RSA key file named "`mail-keys`", that we have created in step 1 above. These files will look like this:

RSA keys for gateway `deep` (`deep-keys`):

```
[root@deep /]# cd /
[root@deep /]# vi deep-keys

    # 1024 bits, Fri Feb  4 05:05:19 2000
    # for signatures only, UNSAFE FOR ENCRYPTION
    #pubkey=0x010395daee1be05f3038ae529ef2668afd79f5ff1b16203c9ceaef801cea9cb74
bcfb51a6ecc08890d3eb4b5470c0fc35465c8ba2ce9d1145ff07b5427e04cf4a38ef98a7f29edcb
4d7689f2da7a69199e4318b4c8d0ea25d33e4f084186a2a54f4b4cec12cca1a5deac3b19d561c16
a76bab772888f1fd71aa08f08502a141b611f
    Modulus:
0x95daee1be05f3038ae529ef2668afd79f5ff1b16203c9ceaef801cea9cb74bcfb51a6ecc08890
d3eb4b5470c0fc35465c8ba2ce9d1145ff07b5427e04cf4a38ef98a7f29edcb4d7689f2da7a6919
9e4318b4c8d0ea25d33e4f084186a2a54f4b4cec12cca1a5deac3b19d561c16a76bab772888f1fd
71aa08f08502a141b611f
    PublicExponent: 0x03
    # everything after this point is secret
    PrivateExponent:
0x63e74967eaea2025c98c69f6ef0753a6a3ff6764157dbdf1f50013471324dd352366f48805b0b
37f232384b2b52ce2ee85d173468b62eaa052381a9588a317b3a1324d01a531a41fa7add6c5efbd
d88f4718feed2bc0246be924e81bb90f03e49ceedf7af0dd48f06f265b519600bd082c6e6bd27ea
a71cc0288df1ecc3b062b
    Prime1:
0xc5b471a88b025dd09d4bd7b61840f20d182d9b75bb7c11eb4bd78312209e3aee7ebfe632304db
6df5e211d21af7fee79c5d45546bea3ccc7b744254f6f0b847f
    Prime2:
0xc20a99feeafe79767122409b693be75f15e1aef76d098ab12579624aec708e85e2c5dd62080c3
a64363f2f45b0e96cb4aef8918ca333a326d3f6dc2c72b75361
    Exponent1:
0x83cda11b0756e935be328fcebad5f6b36573bcf927a80bf2328facb6c0697c9eff2a9976cade7
9ea3ec0be1674fff4512e8d8e2f29c2888524d818df9f5d02ff
    Exponent2:
0x815c66a9f1fefba44b6c2b124627ef94b9411f4f9e065c7618fb96dc9da05f03ec83e8ec055d7
c42ced4ca2e75f0f3231f5061086ccd176f37f9e81da1cf8ceb
    Coefficient:
0x10d954c9e2b8d11f4db1b233ef37ff0a3cecfffad89ba5d515449b007803f577e3bd7f0183ced
dfd805466d62f767f3f5a5731a73875d30186520f1753a7e325
```

RSA keys for gateway `mail` (`mail-keys`):

```
[root@mail /]# cd /
[root@mail /]# vi mail-keys

    # 1024 bits, Fri Feb  4 04:46:59 2000
    # for signatures only, UNSAFE FOR ENCRYPTION
    #pubkey=0x01037631b81f00d5e6f888c542d44dbb784cd3646f084ed96f942d341c7c4686c
bd405b805dc728f8697475f11e8b1dd797550153a3f0d4ff0f2b274b70a2ebc88f073748d1c1c88
21dc6be6a2f0064f3be7f8e4549f8ab9af64944f829b014788dd202cf7d2e320cab666f5e7a197e
64efe0bfee94e92ce4dad82d5230c57b89edf
    Modulus:
0x7631b81f00d5e6f888c542d44dbb784cd3646f084ed96f942d341c7c4686cbd405b805dc728f8
697475f11e8b1dd797550153a3f0d4ff0f2b274b70a2ebc88f073748d1c1c8821dc6be6a2f0064f
3be7f8e4549f8ab9af64944f829b014788dd202cf7d2e320cab666f5e7a197e64efe0bfee94e92c
e4dad82d5230c57b89edf
    PublicExponent: 0x03
    # everything after this point is secret
    PrivateExponent:
0x4ecbd014ab3944a5b08381e2de7cfadde242f4b03490f50d737812fd8459dd3803d003e84c5fa
f0f84ea0bf07693a64e35637c2a08dff5f721a324b1747db09f62c871d5e11711251b845ae76753
d4ef967c494b0def4f5d0762f65da603bc04c41b4c6cab4c413a72c633b608267ae2889c162a3d5
bc07ee083b1c6e038400b
```

```
    Prime1:
0xc7f7cc8feaaac65039c39333b878bffd8f95b0dc22995c553402a5b287f341012253e9f25b839
83c936f6ca512926bebee3d5403bf9f4557206c6bbfd9aac899
    Prime2:
0x975015cb603ac1d488dc876132d8bc83079435d2d3395c03d5386b5c004eadd4d7b01b3d86aad
0a2275d2d6b791a2abe50d7740b7725679811a32ca22db97637
    Exponent1:
0x854fddb5471c84357bd7b777d0507ffe5fb92092c1bb92e37801c3cc5aa22b5616e29bf6e7ad1
028624a486e0c619d47f428e2ad2a6a2e3a159d9d2a911c85bb
    Exponent2:
0x64e00e87957c81385b3daf9621e5d302050d7937377b92ad38d04792aadf1e8de52012290471e
06c1a3e1e47a61171d435e4f807a4c39a6561177316c9264ecf
    Coefficient:
0x6f087591becddc210c2ee0480e30beeb25615a3615203cd3cef65e5a1d476fd9602ca0ef10d9b
858edb22db42c975fb71883a470b43433a7be57df7ace4a0a3f
```

Extract and copy the public RSA key files of `deep` and `mail` to your `ipsec.conf` files as shown below. You can locate the line related to the public key by a sentence beginning with the commented-out: "`#pubkey=`" line.

```
# sample connection
conn deep-mail
    left=208.164.186.1
    leftsubnet=192.168.1.0/24
    leftnexthop=205.151.222.250
    right=208.164.186.2
    rightsubnet=192.168.1.0/24
    rightnexthop=205.151.222.251
    keyingtries=0
    auth=ah
    authby=rsasig
    leftrsasigkey=0x010395daee1be05f3038ae529ef2668afd79f5ff1b16203c9ceaef801ce
    a9cb74bcfb51a6ecc08890d3eb4b5470c0fc35465c8ba2ce9d1145ff07b5427e04cf4a38ef9
    8a7f29edcb4d7689f2da7a69199e4318b4c8d0ea25d33e4f084186a2a54f4b4cec12cca1a5d
    eac3b19d561c16a76bab772888f1fd71aa08f08502a141b611f
    rightrsasigkey=0x01037631b81f00d5e6f888c542d44dbb784cd3646f084ed96f942d341c
    7c4686cbd405b805dc728f8697475f11e8b1dd797550153a3f0d4ff0f2b274b70a2ebc88f07
    3748d1c1c8821dc6be6a2f0064f3be7f8e4549f8ab9af64944f829b014788dd202cf7d2e320
    cab666f5e7a197e64efe0bfee94e92ce4dad82d5230c57b89edf
    auto=start
```

NOTE: Don't forget that, in this example, the "`leftrsasigkey=`" parameter contains the public key of `deep` and the "`rightrsasigkey=`" parameter contains the public key of `mail`.

Step 3
Modify your `/etc/ipsec.secrets` files to use RSA private keys in *each* gateway:

Edit your original **`ipsec.secrets`** file (`vi /etc/ipsec.secrets`) and add the RSA private key for authentication on both gateways:

The `ipsec.secrets` file for gateway `deep`:

```
[root@deep /]# vi /etc/ipsec.secrets

208.164.186.1 208.164.186.2
"0x9748cc31_2e99194f_d230589b_cd846b57_dc070b01_74b66f34_19c40a1a_804906ed"
```

You must change your original `ipsec.secrets` file as shown above to look like the following on both gateways. It is important to note that the private keys are not the same on both gateways, `deep` and `mail`. The private key for `deep` comes from the RSA key file "`deep-keys`", while the private key for mail comes from the RSA key file "`mail-keys`":

```
208.164.186.1 208.164.186.2: RSA {
        Modulus:
0x95daee1be05f3038ae529ef2668afd79f5ff1b16203c9ceaef801cea9cb74bcfb51a6ecc08890
d3eb4b5470c0fc35465c8ba2ce9d1145ff07b5427e04cf4a38ef98a7f29edcb4d7689f2da7a6919
9e4318b4c8d0ea25d33e4f084186a2a54f4b4cec12cca1a5deac3b19d561c16a76bab772888f1fd
71aa08f08502a141b611f
        PublicExponent: 0x03
        # everything after this point is secret
        PrivateExponent:
0x63e74967eaea2025c98c69f6ef0753a6a3ff6764157dbdf1f50013471324dd352366f48805b0b
37f232384b2b52ce2ee85d173468b62eaa052381a9588a317b3a1324d01a531a41fa7add6c5efbd
d88f4718feed2bc0246be924e81bb90f03e49ceedf7af0dd48f06f265b519600bd082c6e6bd27ea
a71cc0288df1ecc3b062b
        Prime1:
0xc5b471a88b025dd09d4bd7b61840f20d182d9b75bb7c11eb4bd78312209e3aee7ebfe632304db
6df5e211d21af7fee79c5d45546bea3ccc7b744254f6f0b847f
        Prime2:
0xc20a99feeafe79767122409b693be75f15e1aef76d098ab12579624aec708e85e2c5dd62080c3
a64363f2f45b0e96cb4aef8918ca333a326d3f6dc2c72b75361
        Exponent1:
0x83cda11b0756e935be328fcebad5f6b36573bcf927a80bf2328facb6c0697c9eff2a9976cade7
9ea3ec0be1674fff4512e8d8e2f29c2888524d818df9f5d02ff
        Exponent2:
0x815c66a9f1fefba44b6c2b124627ef94b9411f4f9e065c7618fb96dc9da05f03ec83e8ec055d7
c42ced4ca2e75f0f3231f5061086ccd176f37f9e81da1cf8ceb
        Coefficient:
0x10d954c9e2b8d11f4db1b233ef37ff0a3cecfffad89ba5d515449b007803f577e3bd7f0183ced
dfd805466d62f767f3f5a5731a73875d30186520f1753a7e325
    }
```

The `ipsec.secrets` file for gateway `mail`:

```
[root@mail /]# vi /etc/ipsec.secrets

208.164.186.1 208.164.186.2: RSA {
        Modulus:
0x95daee1be05f3038ae529ef2668afd79f5ff1b16203c9ceaef801cea9cb74bcfb51a6ecc08890
d3eb4b5470c0fc35465c8ba2ce9d1145ff07b5427e04cf4a38ef98a7f29edcb4d7689f2da7a6919
9e4318b4c8d0ea25d33e4f084186a2a54f4b4cec12cca1a5deac3b19d561c16a76bab772888f1fd
71aa08f08502a141b611f
        PublicExponent: 0x03
        # everything after this point is secret
        PrivateExponent:
0x63e74967eaea2025c98c69f6ef0753a6a3ff6764157dbdf1f50013471324dd352366f48805b0b
37f232384b2b52ce2ee85d173468b62eaa052381a9588a317b3a1324d01a531a41fa7add6c5efbd
d88f4718feed2bc0246be924e81bb90f03e49ceedf7af0dd48f06f265b519600bd082c6e6bd27ea
a71cc0288df1ecc3b062b
        Prime1:
0xc5b471a88b025dd09d4bd7b61840f20d182d9b75bb7c11eb4bd78312209e3aee7ebfe632304db
6df5e211d21af7fee79c5d45546bea3ccc7b744254f6f0b847f
        Prime2:
0xc20a99feeafe79767122409b693be75f15e1aef76d098ab12579624aec708e85e2c5dd62080c3
a64363f2f45b0e96cb4aef8918ca333a326d3f6dc2c72b75361
        Exponent1:
0x83cda11b0756e935be328fcebad5f6b36573bcf927a80bf2328facb6c0697c9eff2a9976cade7
9ea3ec0be1674fff4512e8d8e2f29c2888524d818df9f5d02ff
```

```
        Exponent2:
0x815c66a9f1fefba44b6c2b124627ef94b9411f4f9e065c7618fb96dc9da05f03ec83e8ec055d7
c42ced4ca2e75f0f3231f5061086ccd176f37f9e81da1cf8ceb
        Coefficient:
0x10d954c9e2b8d11f4db1b233ef37ff0a3cecfffad89ba5d515449b007803f577e3bd7f0183ced
dfd805466d62f767f3f5a5731a73875d30186520f1753a7e325
    }
```

Authentication by RSA Signatures requires that each host have its own private key. The key part of an entry may start with a token indicating the kind of key. "RSA" signifies RSA private key and "PSK" (which is the default) signifies PreShared Key. Since "PSK" is the default, we must specify "RSA", so that we'll be able to use RSA private keys in this file (ipsec.secrets). The super-user "root" should own the file ipsec.secrets, and its permissions should be set to block all access by others.

Requiring network setup for IPSec

There are some considerations you must ensure are correct before running FreeS/WAN software. These considerations are important if you don't want to receive error messages during start up of your VPN. The following are the steps to follow:

Step1
You will need to enable TCP/IP forwarding on the both gateway servers. In Linux, this is accomplished by adding the following line:

- To enable IPv4 forwarding on your Linux system, edit the /etc/sysctl.conf file (vi /etc/sysctl.conf) and add the following line:

```
# Enable/Disable packet forwarding
net.ipv4.ip_forward = 1
```

You must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

Step 2
Recall that automatically keyed connections use keys automatically generated by the Pluto key negotiation daemon. The pluto daemon will start up, try to connect to the Pluto daemon at the other end of the tunnel, and establish a connection. For this reason, an IPSEC gateway should have packet filters rules (in the firewall script file) permitting the following protocols to traverse the gateway when talking to other IPSEC gateway:

- ✓ UDP port 500 for IKE implemented by the Pluto daemon
- ✓ Protocol 50 for ESP encryption and/or authentication
- ✓ Protocol 51 for AH packet-level authentication

See the GIPTables chapter in this book and the GIPTables manual for the correct rules to add to your firewall on both gateway machines to allow IPSEC packets to traverse the remote network gateway to your network gateway and vice versa.

Step 3
The rp_filter subsystem (related to IP spoofing protection) must be turned off on both gateways for IPSEC to work properly. This is accomplished by checking if the value 0 (off) is set in the /proc/sys/net/ipv4/conf/ipsec0/rp_filter and /proc/sys/net/ipv4/conf/eth0/rp_filter files respectively:

- To check if the value 0 (off) is set in the rp_filter files, use the commands:
```
[root@deep /]# cat /proc/sys/net/ipv4/conf/ipsec0/rp_filter
0
[root@deep /]# cat /proc/sys/net/ipv4/conf/eth0/rp_filter
0
```

NOTE: The subdirectory "ipsec0" in our example will be created only after the reboot of your system. So you may check the value of the "rp_filter" file in the "ipsec0" directory after your system has been restarted.

- To set the value 0 (off) in the both rp_filter files manually, use the commands:
```
[root@deep /]# echo 0 > /proc/sys/net/ipv4/conf/ipsec0/rp_filter
[root@deep /]# echo 0 > /proc/sys/net/ipv4/conf/eth0/rp_filter
```

Also you can put lines like the following in your firewall script files /etc/rc.d/init.d/iptables on both gateways to automatically set these values to 0 (off) and avoid making them manually:

```
# Disable IP spoofing protection to allow IPSEC to work properly
echo 0 > /proc/sys/net/ipv4/conf/ipsec0/rp_filter
echo 0 > /proc/sys/net/ipv4/conf/eth0/rp_filter
```

NOTE: In the example of the firewall script file above, we assume that eth0 is the interface you use for your connection. Of course if you use eth1 you must change eth0 to eth1, and so on.

If you forget this step you will receive error messages on your terminal such as the following during the start up of FreeSWAN IPSEC:

```
ipsec_setup: WARNING: ipsec0 has route filtering turned on, KLIPS may not work
ipsec_setup:  (/proc/sys/net/ipv4/conf/ipsec0/rp_filter = `1', should be 0)
ipsec_setup: WARNING: eth0 has route filtering turned on, KLIPS may not work
ipsec_setup:  (/proc/sys/net/ipv4/conf/eth0/rp_filter = `1', should be 0)
```

Testing the FreeS/WAN installation

- Reboot the both gateways to get FreeS/WAN started.

- Examine the /var/log/messages file for any signs of trouble. If all goes well you should see something like this in the /var/log/messages file:

```
Feb  2 05:22:35 deep ipsec_setup: Starting FreeS/WAN IPSEC
snap2000jan31b...
Feb  2 05:22:35 deep ipsec_setup: KLIPS debug `none'
Feb  2 05:22:35 deep ipsec_setup: KLIPS ipsec0 on eth0
192.168.1.1/255.255.255.0 broadcast 192.168.1.255
Feb  2 05:22:36 deep ipsec_setup: Disabling core dumps:
Feb  2 05:22:36 deep ipsec_setup: Starting Pluto (debug `none'):
Feb  2 05:22:37 deep ipsec_setup: Loading Pluto database `deep-mail':
Feb  2 05:22:37 deep ipsec_setup: Enabling Pluto negotiation:
Feb  2 05:22:37 deep ipsec_setup: Routing for Pluto conns `deep-mail':
Feb  2 05:22:37 deep ipsec_setup: Initiating Pluto tunnel `deep-mail':
Feb  2 05:22:39 deep ipsec_setup: 102 "deep-mail" #1: STATE_MAIN_I1:
initiate
Feb  2 05:22:39 deep ipsec_setup: 104 "deep-mail" #1: STATE_MAIN_I2: from
STATE_MAIN_I1; sent MI2, expecting MR2
Feb  2 05:22:39 deep ipsec_setup: 106 "deep-mail" #1: STATE_MAIN_I3: from
STATE_MAIN_I2; sent MI3, expecting MR3
Feb  2 05:22:39 deep ipsec_setup: 004 "deep-mail" #1: STATE_MAIN_I4: SA
established
Feb  2 05:22:39 deep ipsec_setup: 110 "deep-mail" #2: STATE_QUICK_I1:
initiate
Feb  2 05:22:39 deep ipsec_setup: 004 "deep-mail" #2: STATE_QUICK_I2: SA
established
Feb  2 05:22:39 deep ipsec_setup: ...FreeS/WAN IPSEC started
```

- Examine the /var/log/secure file for any signs of trouble. If all goes well you should see something like the following:

```
Feb 21 14:45:42 deep Pluto[432]: Starting Pluto (FreeS/WAN Version 1.3)
Feb 21 14:45:43 deep Pluto[432]: added connection description "deep-mail"
Feb 21 14:45:43 deep Pluto[432]: listening for IKE messages
Feb 21 14:45:43 deep Pluto[432]: adding interface ipsec0/eth0 192.168.1.1
Feb 21 14:45:43 deep Pluto[432]: loading secrets from
"/etc/ipsec.secrets"
Feb 21 14:45:43 deep Pluto[432]: "deep-mail" #1: initiating Main Mode
Feb 21 14:45:44 deep Pluto[432]: "deep-mail" #1: ISAKMP SA established
Feb 21 14:45:44 deep Pluto[432]: "deep-mail" #2: initiating Quick Mode
POLICY_RSASIG+POLICY_ENCRYPT+POLICY_AUTHENTICATE+POLICY_TUNNEL+POLICY_PFS
Feb 21 14:45:46 deep Pluto[432]: "deep-mail" #2: sent QI2, IPsec SA
established
Feb 21 14:45:47 deep Pluto[432]: "deep-mail" #3: responding to Main Mode
Feb 21 14:45:49 deep Pluto[432]: "deep-mail" #3: sent MR3, ISAKMP SA
established
Feb 21 14:45:49 deep Pluto[432]: "deep-mail" #4: responding to Quick Mode
Feb 21 14:45:50 deep Pluto[432]: "deep-mail" #4: IPsec SA established
```

- On both gateways, the following entries should now exist in the `/proc/net/` directory:

```
[root@deep /]# ls -l /proc/net/ipsec_*
-r--r--r--   1 root     root  0 Feb  2 05:30 /proc/net/ipsec_eroute
-r--r--r--   1 root     root  0 Feb  2 05:30 /proc/net/ipsec_klipsdebug
-r--r--r--   1 root     root  0 Feb  2 05:30 /proc/net/ipsec_spi
-r--r--r--   1 root     root  0 Feb  2 05:30 /proc/net/ipsec_spigrp
-r--r--r--   1 root     root  0 Feb  2 05:30 /proc/net/ipsec_spinew
-r--r--r--   1 root     root  0 Feb  2 05:30 /proc/net/ipsec_tncfg
-r--r--r--   1 root     root  0 Feb  2 05:30 /proc/net/ipsec_version
```

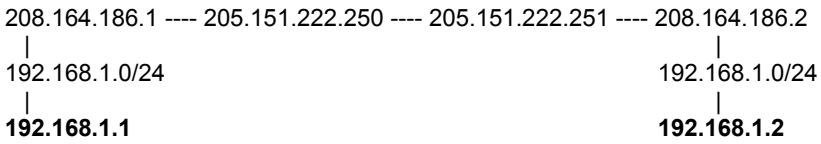
- The `IPSEC` interfaces should be attached on top of the specified physical interfaces. Confirm that with:

```
[root@deep /]# cat /proc/net/ipsec_tncfg
ipsec0 -> eth0 mtu=16260 -> 1500
ipsec1 -> NULL mtu=0 -> 0
ipsec2 -> NULL mtu=0 -> 0
ipsec3 -> NULL mtu=0 -> 0
```

- Now execute the following command to show minimal debugging information and see if the output looks something like this:

```
[root@deep /]# ipsec look
deep.openna.com Fri Feb  4 17:25:17 EST 2000
============-============
192.168.1.1/32     -> 192.168.1.2/32     => tun0x106@192.168.1.2
esp0x4450894d@192.168.1.2 ah0x4450894c@192.168.1.2
------------=------------
ah0x3350f551@192.168.1.1 AH_HMAC_MD5: dir=in ooowin=32 seq=115
bit=0xffffffff alen=128 aklen=16
life(c,s,h)=bytes(16140,0,0)add(51656,0,0)use(54068,0,0)packets(115,0,0)
idle=499
ah0x4450894c@192.168.1.2 AH_HMAC_MD5: dir=out ooowin=32 seq=2828 alen=128
aklen=16
life(c,s,h)=bytes(449488,0,0)add(51656,0,0)use(51656,0,0)packets(2828,0,0
) idle=6
esp0x3350f552@192.168.1.1 ESP_3DES: dir=in ooowin=32 seq=115
bit=0xffffffff eklen=24
life(c,s,h)=bytes(13380,0,0)add(51656,0,0)use(54068,0,0)packets(115,0,0)
idle=499
esp0x4450894d@192.168.1.2 ESP_3DES: dir=out ooowin=32 seq=2828 eklen=24
life(c,s,h)=bytes(381616,0,0)add(51656,0,0)use(51656,0,0)packets(2828,0,0
) idle=6
tun0x105@192.168.1.1 IPIP: dir=in 192.168.1.2 -> 192.168.1.1
life(c,s,h)=add(51656,0,0)
tun0x106@192.168.1.2 IPIP: dir=out 192.168.1.1 -> 192.168.1.2
life(c,s,h)=bytes(327581,0,0)add(51656,0,0)use(51656,0,0)packets(2828,0,0
) idle=6
192.168.1.0     0.0.0.0  255.255.255.0   U   0 0 0 eth0
192.168.1.0     0.0.0.0  255.255.255.0   U   0 0 0 ipsec0
192.168.1.1     0.0.0.0  255.255.255.255 UH  0 0 0 eth0
192.168.1.2     192.168.1.2  255.255.255.255 UGH 0 0 0 ipsec0
Destination     Gateway  Genmask Flags   MSS Window  irtt Iface
```

- Try pinging **192.168.1.2** from the **192.168.1.1** client. If this works then you have set it up correctly. If it does not work check your network to make sure 208.164.186.1 can reach 208.164.186.2, and that TCP-IP forwarding is enabled, and make sure that no firewall rules are blocking the packets, or trying to masquerade them before the rules allowing IPSec related traffic. For this test to work, it is important to use pings that go from one subnet to the other.



A last note about testing the installation of FreeSWAN IPSEC, if you encounter a problem that you are unable to resolve, you can use the following command to view a collection of debugging information (contents of files, selections from logs, etc.) related to the IPSEC encryption/authentication system that you should send to the Linux-IPSEC Mailing List (linux-ipsec@clinet.fi) to help you.

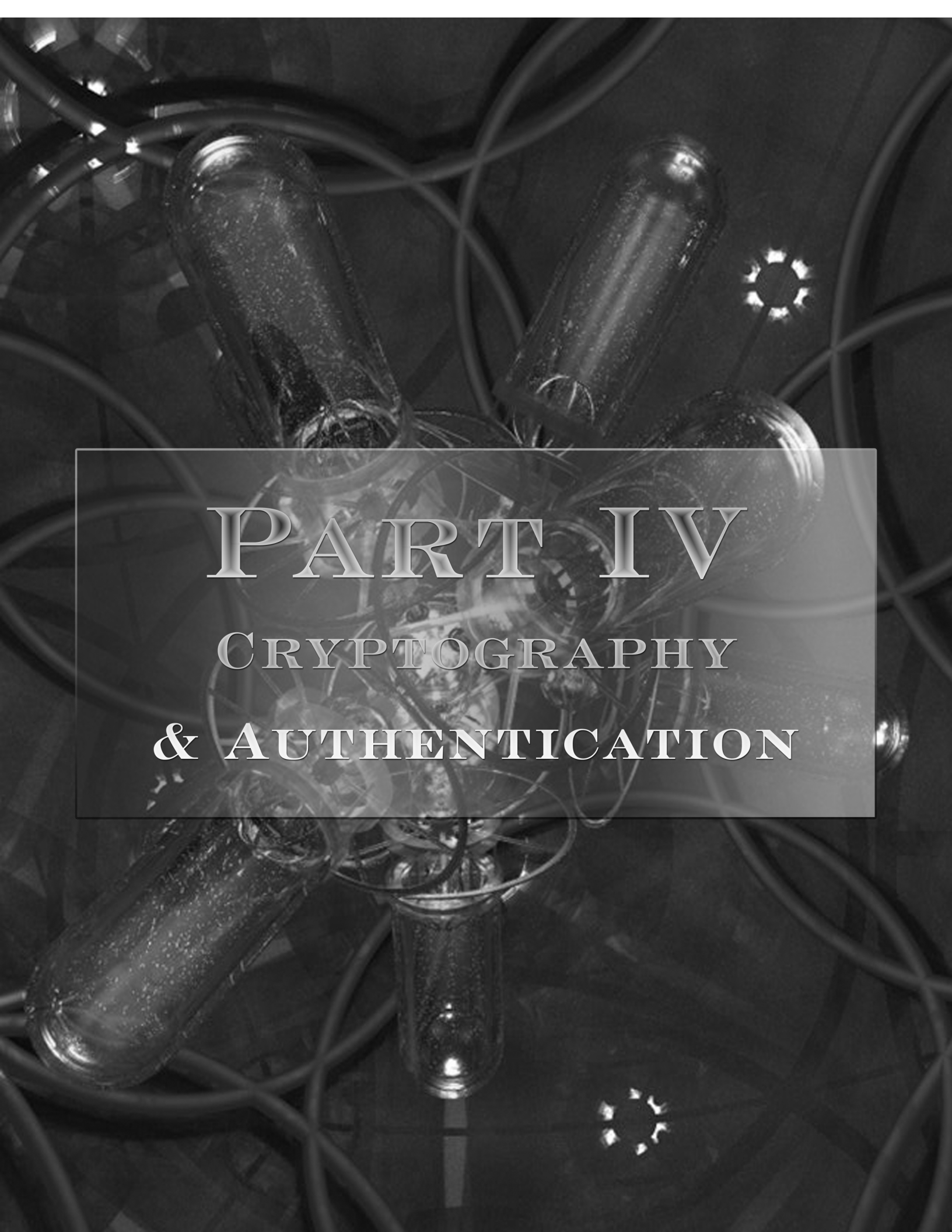
- Use the following command to make an output of a collection of debugging information:
 [root@deep /]# **ipsec barf > result**

This command is primarily provided as a convenience for remote debugging; A single command which packages up (and labels) all information that might be relevant to diagnosing a problem in IPSEC.

Further documentation

For more details, there are several manual pages about FreeS/WAN that you could read:

$ man ipsec (8)	- invoke IPSEC utilities.
$ man ipsec atoaddr, addrtoa (3)	- convert Internet addresses to and from ASCII.
$ man ipsec atoasr (3)	- convert ASCII to Internet address, subnet, or range.
$ man ipsec atobytes, bytestoa (3)	- convert binary data bytes from and to ASCII formats.
$ man ipsec atodata, datatoa (3)	- convert binary data from and to ASCII formats.
$ man ipsec atosa, satoa (3)	- convert IPSEC Security Association IDs to and from ASCII.
$ man ipsec atosubnet, subnettoa (3)	- convert subnet/mask ASCII form to and from addresses.
$ man ipsec atoul, ultoa (3)	- convert unsigned-long numbers to and from ASCII.
$ man ipsec auto (8)	- control automatically-keyed IPSEC connections.
$ man ipsec barf (8)	- spew out collected IPSEC debugging information.
$ man ipsec bitstomask (3)	- convert bit count to Internet subnet mask.
$ man ipsec eroute (8)	- manipulate IPSEC extended routing tables.
$ man ipsec goodmask (3)	- is this Internet subnet mask a valid one?
$ man ipsec hostof (3)	- given Internet address and subnet mask, return host part.
$ man ipsec klipsdebug (8)	- set Klips (kernel IPSEC support) debug features and level.
$ man ipsec look (8)	- show minimal debugging information.
$ man ipsec manual (8)	- take manually-keyed IPSEC connections up and down.
$ man ipsec masktobits (3)	- convert Internet subnet mask to bit count.
$ man ipsec optionsfrom (3)	- read additional ``command-line'' options from file.
$ man ipsec pluto (8)	- IPsec IKE keying daemon.
$ man ipsec ranbits (8)	- generate random bits in ASCII form.
$ man ipsec rangetoa (3)	- convert Internet address range to ASCII.
$ man ipsec rsasigkey (8)	- generate RSA signature key.
$ man ipsec setup (8)	- control IPSEC subsystem.
$ man ipsec spi (8)	- manage IPSEC Security Associations.
$ man ipsec spigrp (8)	- group/ungroup IPSEC Security Associations.
$ man ipsec subnetof (3)	- given Internet address and subnet mask, return subnet number.
$ man ipsec tncfg (8)	- associate IPSEC virtual interface with real interface.
$ man ipsec whack (8)	- control interface for IPSEC keying daemon.
$ man ipsec.conf (5)	- IPSEC configuration and connections.
$ man ipsec.secrets (5)	- secrets for IKE/IPsec authentication.



Part IV

Cryptography & Authentication

CHAPTER 14

GnuPG

IN THIS CHAPTER

1. Compiling - Optimizing & Installing `GnuPG`
2. Using `GnuPG` under Linux terminal

Linux GnuPG

Abstract

At this point we are ready to compile, configure, optimize and install software on our Linux server. Yes it is time, and we will begin our adventure with the powerful and easy to install GnuPG tool. Why do we choose to begin with GnuPG? The answer is simple, we are playing with a highly secured server and the first action to take each time we want to install some new software on this secured machine is to be absolutely sure that the software in question comes from a trusted source and is unmodified. With the GnuPG tool we can verify the supplied signature and be sure that the software is original. So it is recommended that this program is installed before any others.

Encryption of data sources is an invaluable feature that gives us a high degree of confidentiality for our work. A tool like GnuPG does much more than just encryption of mail messages. It can be used for all kinds of data encryption, and its utilization is only limited by the imagination.

GnuPG is GNU's tool for secure data communication and storage. It can be used to encrypt data and to create digital signatures. It includes an advanced key management facility and is compliant with the proposed OpenPGP Internet standard as described in RFC2440. Because GnuPG does not use any patented algorithm it is not compatible with PGP2 versions.

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest GnuPG version number is 1.0.7

The procedures given in this chapter are likely to work on all Linux platforms, but we have only tested it on OpenNA Linux and Red Hat Linux.

Packages

Please check http://www.gnupg.org/ regularly for the latest status. We chose to install from source because it provides the facility to fine tune the installation.

Source code is available from:
GnuPG Homepage: http://www.gnupg.org/
GnuPG FTP Site: 217.69.76.44
You must be sure to download: gnupg-1.0.7.tar.gz

Prerequisites

GnuPG requires that the listed software below be already installed on your system to be able to compile successfully. If this is not the case, you must install it from your Linux CD-ROM or source archive files. Please make sure you have this program installed on your machine before you proceed with this chapter.

- ✓ gettext is required to run GnuPG on your system.
- ✓ python-1.5 is required to run GnuPG on your system.
- ✓ expat is required to run GnuPG on your system.
- ✓ gmp is required to run GnuPG on your system.

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all the files installed on the system if you want to update the package in the future. To solve this problem, it's a good idea to make a list of files on the system before you install GnuPG, and then one afterwards, and then compare them using the **diff** utility to find out what files were placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep root]# find /* > GnuPG1
  ```
- And the following one after you install the software:
  ```
  [root@deep root]# find /* > GnuPG2
  ```
- Then use the following command to get a list of what changed:
  ```
  [root@deep root]# diff GnuPG1 GnuPG2 > GnuPG-Installed
  ```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. In our example above, we use the /root directory of the system to store all the generated file lists.

Compiling - Optimizing & Installing GnuPG

Below are the required steps that you must make to configure, compile and optimize the GnuPG software before installing it into your Linux system. First off, we install the program as user 'root' so as to avoid authorization problems.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- This can be done with the following commands:
  ```
  [root@deep /]# cp gnupg-version.tar.gz /var/tmp/
  [root@deep /]# cd /var/tmp/
  [root@deep tmp]# tar xzpf gnupg-version.tar.gz
  ```

Step 2
In order to check that the version of GnuPG, which you are going to install, is an original and unmodified one, use the commands described below and check the supplied signature. Since we don't have GnuPG already installed in the system, we have to verify the MD5 checksum of the program.

- To verify the MD5 checksum of GnuPG, use the following command:
```
[root@deep /]# md5sum gnupg-version.tar.gz
```

This should yield an output similar to this:
```
d8b36d4dfd213a1a1027b1877acbc897  gnupg-1.0.7.tar.gz
```

Now check that this checksum is exactly the same as the one published on the GnuPG website at the following URL: http://www.gnupg.org/download.html

Step 3
Next, move into the newly created GnuPG source directory and perform the following steps to configure and optimize the software for your system.

- To move into the newly created GnuPG directory use the following command:
```
[root@deep tmp]# cd gnupg-1.0.7/
```

- To configure and optimize GnuPG use the following compilation lines:
```
CFLAGS="-O2 -march=i686 -funroll-loops"; export CFLAGS
./configure \
--prefix=/usr \
--mandir=/usr/share/man \
--infodir=/usr/share/info \
--enable-shared \
--disable-nls
```

WARNING: Pay special attention to the compile CFLAGS line above. We optimize GnuPG for an i686 CPU architecture with the parameter "-march=i686". Please don't forget to adjust the CFLAGS line to reflect your own system.

Step 4
Now, we must make a list of all files on the system before installing the software, and one afterwards, then compare them using the **diff** utility to find out what files are placed where and finally we install the GnuPG software:

```
[root@deep gnupg-1.0.7]# make
[root@deep gnupg-1.0.7]# make check
[root@deep gnupg-1.0.7]# cd
[root@deep root]# find /* > GnuPG1
[root@deep root]# cd /var/tmp/gnupg-1.0.7/
[root@deep gnupg-1.0.7]# make install
[root@deep gnupg-1.0.7]# strip /usr/bin/gpg
[root@deep gnupg-1.0.7]# strip /usr/bin/gpgv
[root@deep gnupg-1.0.7]# cd
[root@deep root]# find /* > GnuPG2
[root@deep root]# diff GnuPG1 GnuPG2 > GnuPG-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

Step 5
Once the configuration, optimization, compilation, and installation of the GnuPG software have been accomplished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete GnuPG and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf gnupg-version/
[root@deep tmp]# rm -f gnupg-version.tar.gz
```

The rm command as used above will remove all the source files we have used to compile and install GnuPG. It will also remove the GnuPG compressed archive from the /var/tmp/ directory.

Using GnuPG under Linux terminal

Here we show you how to use GnuPG using a terminal to manage GPG keys. The commands listed below are ones that we use often, but many more exist. Check the manual page gpg (1) for more information.

Creating a key-pair:

First of all, we must create a new key-pair (public and private) if this is a first use of the GnuPG software to be able to use its encryption features.

Step 1
The "--gen-key" option of GnuPG is used to generate a new (public and private) key, we have to use it every time we need to create a new GnuPG key on the system. When we issue this command for the first time, GnuPG will create the required directory and options file for us.

- To create a new key-pair, use the following command:

```
[root@deep /]# gpg --gen-key
gpg (GnuPG) 1.0.7; Copyright (C) 2000 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.

gpg: /root/.gnupg: directory created
gpg: /root/.gnupg/options: new options file created
gpg: you have to start GnuPG again, so it can read the new options file
```

Step 2
Once the command has been executed, we have to run it again for a second time to create our public and private keys, because on first utilization, it just creates the required directory and options file for us. Therefore, it will now create the keys.

- We start GnuPG again with the same command:

```
[root@deep /]# gpg --gen-key
gpg (GnuPG) 1.0.7; Copyright (C) 2002 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.

gpg: keyring `/root/.gnupg/secring.gpg' created
gpg: keyring `/root/.gnupg/pubring.gpg' created
Please select what kind of key you want:
   (1) DSA and ElGamal (default)
   (2) DSA (sign only)
   (4) ElGamal (sign and encrypt)
   (5) RSA (sign only)
Your selection? 1
DSA keypair will have 1024 bits.
About to generate a new ELG-E keypair.
              minimum keysize is  768 bits
              default keysize is 1024 bits
    highest suggested keysize is 2048 bits
What keysize do you want? (1024) Press Enter
Requested keysize is 1024 bits
Please specify how long the key should be valid.
         0 = key does not expire
      <n>  = key expires in n days
      <n>w = key expires in n weeks
      <n>m = key expires in n months
      <n>y = key expires in n years
Key is valid for? (0) Press Enter
Key does not expire at all
Is this correct (y/n)? y

You need a User-ID to identify your key; the software constructs the user
id from Real Name, Comment and Email Address in this form:
    "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Real name: Gerhard Mourani
Email address: gmourani@openna.com
Comment: Press Enter
You selected this USER-ID:
    "Gerhard Mourani <gmourani@openna.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? o
You need a Passphrase to protect your secret key.

Enter passphrase: mypassphrase
Repeat passphrase: mypassphrase

We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
++++++++++++++++++++.+++++.+++++.+++++++++++++++.+++++++++++++++..+++++++
++++++++...+++++++++++++++.+++++.++++++++++++++++++++++++++++++++++++++++
..+++++
```

```
<+++++..>+++++.................<..+++++.................................
.........+++++^^^^
public and secret key created and signed.
```

NOTE: A new key-pair is created (secret and public key) in the "root" home directory ~/root under the .gnupg subdirectory because we issued this GnuPG command as user "root". If you run the above command under other user into the system, then the generated keys will be located under its home directory on the server.

Exporting GPG key/s for a user:

Once your own key-pair is created, you can expand your horizons by exporting and distributing your public key over the world (NEVER export you private key). This can be done by publishing it on your homepage, through an available key server on the Internet, or any other available method. GnuPG has some useful options to help you publish your public key.

Step 1
First off, we have to extract our public key in ASCII text to be able to distribute it. ASCII text is a good format to use because it allows people to get it easily. In this way, anyone can just cut and past your public key and use it when they want to securely communicate with you.

- To extract your public key in ASCII armored output, use the following command:
  ```
  [root@deep /]# gpg --export -ao UID
  ```

 As an example:
  ```
  [root@deep /]# gpg --export -ao Gerhard Mourani
  ```

Where "--export" is for extracting Public-key from your pubring encrypted file, "a" is to create ASCII armored output that you can mail, publish or put it on a web page, "o" to put the result in a file and UID represents the user key you want to export, which is in our example the user "Gerhard Mourani" key that we have create previously.

Step 2
Once your public key has been extracted, the resulting output will be a file called "Gerhard" under the directory where you are issuing the above command, representing the First name of the user key to extract. In our example, the file is called "Gerhard" because it is the name of the key we want to export in ASCII text format. Note that the file name will be different for your public ASCII text format key.

- Edit your public key in ASCII armored output format, and distribute it:
  ```
  [root@deep /]#  vi Gerhard
  -----BEGIN PGP PUBLIC KEY BLOCK-----
  Version: GnuPG v1.0.7 (GNU/Linux)
  Comment: For info see http://www.gnupg.org

  mQGiBDzGNQcRBAC+1NrjFMCEtyjcv5lhtFNMLHEQ0VdHObv0CMUdCkiDslJ9QT9v
  MtVG1d4r3+0RJan23Z+8fc11E7Q0wRjRO13efRGEbxaIushhRc/p11LsEubWMWC7
  E1UCsMmniScEdoZLSq9/myjj7IJqAavgL0a7/VkVHjrX1j/pTTK1wUUsRwCgy0jp
  0JzY1+dIK4ElfGxAQ7oHop8D/03MkyVhUZh9asLW4tyGlmMN8exqfRoMdeSv0jnz
  ftAAZ71sn8jDdviccvaJvj2eTdZ7J43BIhxALJZ8KMQdEDWQnW62FfV9uGWcB5HL
  c869XOD0so9LOJGsgF1XpnMKQhTRXXEIuN0THpGDSLdQtXelBzIusQuSmNBrx7A0
  6/5xA/0W3H2NYzvMWnTuENpHUQR8KtIARcmis4bGIH/fEiPQyR7YWIAs9sPOE5Yr
  3cQuUpZ3nwGcZ5CGOKm0qRBkhMI49SO25gsoaRVVatNZ1v1o07AaNDimmvE0hhO3
  +/LTv9cJYMdm4ijp+XOhssO4zctgdg0bHISsTWqB1AJcSsdAirQpR2VyaGFyZCBN
  ```

```
b3VyYW5pIDxzeXNhZG1pbkBkZXYub3Blbm5hLmNvbT6IVwQTEQIAFwUCPMY1BwUL
BwoDBAMVAwIDFgIBAheAAAoJEOTyFOEuU3j3OB8AoJcMlZkGYlHBt013kjg6U7Xt
e7muAJ9LBfIlSHtmR3aZAn/4yekA8jwkrbkBDQQ8xjULEAQAvA7lwVx/AUga4j3d
yo4upmHClk4+rYW9bQQXdMGj9EO2gdrxXzbQ2AlQj0UXgDN8HzXHdcZ4TyGghNVm
zq9k2+Ud4Gx0+q34tJI+ljDM7eGhBZbSMGs7kB75/DKIvqONV2JCYJMutrRQPBF1
8ZRf/FgJEtOcjOHu5UfpMresWXsAAwYEAKj2b7LmSfPpm9X/eTEoHAFbR5WPXkRP
eNUEgN2nk2rzyA+7IL4Sg9OPz31qhKOCh/NhFHKcg5VCS4bG35p78eb9KHr8CO01
+h1lUmqCf+s9UvHLUGJahnfp3lnFul9qBqK9MXvWd2bXfovHzAObC1kWAXuYmfnw
8RxdVSgFD4VyiEYEGBECAAYFAjzGNQsACgkQ5PIU4S5TePeMrwCgslkWPnwc3aTY
xQnMq9ml/PdIhS0An1P917iFxhfP2mneemt4N6ELcF4E
=7bvq
-----END PGP PUBLIC KEY BLOCK-----
```

WARNING: Never export or distribute your private key to the world. I know, this seem to be a stupid warning, but I've been informed that some people do it.

Importing GPG key/s from a user:

When you receive someone's public key (or some trusted third partly keys) you have to add them to your key database in order to be able to use his/her keys for future encryption, verification and authentication. This is often the case, when we install software that has a GPG key available for verification. Therefore, here is what you should do before installing software that has a GPG key to your disposal for authenticity.

Step 1
First off, we have to retrieve the GPG public key of the company, organization, etc that we want to import into our keyring database. In our example, we will retrieve the GPG key that OpenNA uses to sign RPM packages and other software.

This GPG public key is available from: http://www.openna.com/about/openna.asc. Cut and past it into a file called "openna.asc" on your server machine where GnuPG is installed.

Step 2
Now, we have to import the OpenNA GPG public key into our database. This procedure should be done for any GPG public keys that you want to use to verify authenticity of software you want to install on your server. Most organizations have GPG public keys for you to download.

- To import Public Keys to your keyring database, use the following command:
  ```
  [root@deep /]# gpg --import filename
  ```

 As an example:
  ```
  [root@deep /]# gpg --import openna.asc
  gpg: key 3487965A: public key imported
  gpg: /root/.gnupg/trustdb.gpg: trustdb created
  gpg: Total number processed: 1
  gpg:               imported: 1
  ```

The above command will append the new key "filename" into the keyring database and will update all already existing keys. It is important to note that GnuPG does not import keys that are not self-signed (asc).

Signing GPG key/s from a user:

When you import keys into your public keyring database and are sure that the trusted third party is really the person they claim, you can start signing his/her keys. Signing a key certifies that you know the owner of the keys and this leads to the situation where the signature acknowledges that the user ID mentioned in the key is actually the owner of that key.

- To sign the key for company OpenNA that we have added into our keyring database above, use the following command:
  ```
  [root@deep /]# gpg --sign-key UID
  ```

 As an example:
  ```
  [root@deep /]# gpg --sign-key OpenNA

  pub  1024D/3487965A  created: 2001-07-02 expires: never      trust: -/q
  sub  1024g/0146F594  created: 2001-07-02 expires: never
  (1). OpenNA Inc. <noc@openna.com>


  pub  1024D/3487965A  created: 2001-07-02 expires: never      trust: -/q
               Fingerprint: 7A3D 6871 2DF1 9210 8ABE  AF36 D460 86D5 3487
  965A

       OpenNA Inc. <noc@openna.com>

  Are you really sure that you want to sign this key
  with your key: "Gerhard Mourani <gmourani@openna.com>"

  Really sign? y

  You need a passphrase to unlock the secret key for
  user: "Gerhard Mourani <gmourani@openna.com>"
  1024-bit DSA key, ID 2E5378F7, created 2002-04-24

  Enter passphrase:
  ```

WARNING: You should only sign a key as being authentic when you are ABSOLUTELY SURE that the key is really authentic! You should never sign a key based on any kind of assumption.

Checking GPG signature:

We have shown above how to sign a key, now we will explain how people can verify if the signature is really the good one. Once you have extracted your public key and exported it, everyone who knows or gets your public key should be able to check whether encrypted data from you is also really signed by you.

- To check the signature of encrypted data, use the following command:
  ```
  [root@deep /]# gpg --verify Data
  ```

The "--verify" option will check the signature where Data is the encrypted data/file you want to verify.

Encrypting and decrypting GPG files:

After installing, importing, signing and configuring everything in the way that we want, we can start encrypting and decrypting our files, software, etc.

- To encrypt and sign data for the user OpenNA that we have added on our keyring database above, use the following command:

```
[root@deep /]# gpg -sear OpenNA file
```

 As an example:

```
[root@deep /]# gpg -sear OpenNA Message-to-OpenNA.txt

You need a passphrase to unlock the secret key for
user: "Gerhard Mourani <gmourani@openna.com>"
1024-bit DSA key, ID 2E5378F7, created 2002-04-24
Enter passphrase:
```

Of the arguments passed, the "s" is for signing (To avoid the risk that somebody else claims to be you, it is very useful to sign everything you encrypt), "e" for encrypting, "a" to create ASCII armored output (".asc" ready for sending by mail), "r" to encrypt the UID name and "file" is the message you want to encrypt.

- To decrypt data, use the following command:

```
[root@deep /]# gpg -d file
```

 For example:

```
[root@deep /]# gpg -d Message-from-GerhardMourani.asc
You need a passphrase to unlock the secret key for
user: "Gerhard Mourani (Open Network Architecture) <gmourani@openna.com>"
1024-bit DSA key, ID 2E5378F7, created 2002-04-24
Enter passphrase:
```

Where "d" is for decrypting and "file" is the message you want to decrypt. It is important that the public key of the sender of the message we want to decrypt be in our public keyring database or of course nothing will work.

Further documentation

For more details, there are some manual pages about GnuPG that you could read:

```
$ man gpg (1)              - GPG encryption and signing tool.
$ man gpgv (1)             - GPGV signature verification tool.
```

CHAPTER 15

OpenSSL

IN THIS CHAPTER

1. **Compiling - Optimizing & Installing `OpenSSL`**
2. **Configuring `OpenSSL`**
3. **`OpenSSL` Administrative Tools**
4. **Securing `OpenSSL`**

Linux OpenSSL

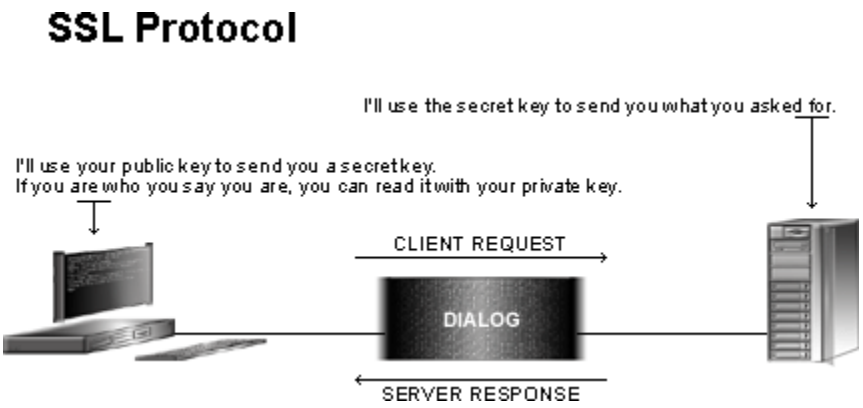
Abstract

The majority of Internet protocols like IMAP, POP, SQL, SMTP, SMB, HTTP, FTP, and LDAP, provide now support for SSL encryption. The big problem in the past was that they asked users to authenticate themselves before allowing access to services, and then they would transmit the users' login ID's and passwords in plain text format over the network, allowing external crackers, using sniffer tools, to get the information and log in into the server themselves.

Encryption mechanisms like SSL ensure safe and secure transactions to eliminate this problem. With this technology, data going over the network is point-to-point encrypted. OpenSSL is a free implementation of this SSL support for all Internet protocols that could run with it (most now do). Once OpenSSL has been installed on your Linux server you can use it as a third party tool to enable SSL functionality with other applications.

The OpenSSL Project is a collaborative effort to develop a robust, commercial-grade, fully featured, and Open Source toolkit implementing the **S**ecure **S**ockets **L**ayer (SSL v2/v3) and **T**ransport **L**ayer **S**ecurity (TLS v1) protocols with full-strength cryptography.

In this chapter, we'll show you how to install OpenSSL for your servers, and how to use it to create certificate keys used by third party software to provide SSL support for, and encryption of, usernames and passwords. Most of the software described in this book needs the presence of OpenSSL on the system to be able to be compiled with SSL support. Therefore, I strongly recommend that you install this encryption software on your Linux system.



Summary of the Cryptographic Thechnology.

Cryptography Advantages

The main advantages gained by using encryption technology are:

Data Confidentiality

When a message is encrypted, an algorithm converts it into enciphered text that hides the meaning of the message, which can then be sent via any public mechanism, and transforms the input plain text. This process involves a secret key that is used to encrypt and later decrypt the data. Without the secret key, the encrypted data is meaningless.

Data Integrity

A cryptographic checksum, called a **M**essage **A**uthentication **C**ode (`MAC`), can be calculated on an arbitrarily user-supplied text to protect the integrity of the data. The results (text and `MAC`) are then sent to the receiver who can verify the trial `MAC` appended to a message by recalculating the `MAC` for the message, using the appropriate secret key and verifying that it matches exactly the trial `MAC`.

Authentication

Personal identification is another use of cryptography, where the user/sender knows a secret, which can serve to authenticate his or her identity.

Electronic Signature

A digital signature assures the sender and receiver that the message is authentic and that only the owner of the key could have generated the digital signature.

Disclaimer

This software package uses strong cryptography, so even if it is created, maintained and distributed from liberal countries in Europe (where it is legal to do this), it falls under certain export/import and/or use restrictions in some other parts of the world.

PLEASE REMEMBER THAT EXPORT/IMPORT AND/OR USE OF STRONG CRYPTOGRAPHY SOFTWARE, PROVIDING CRYPTOGRAPHY HOOKS OR EVEN JUST COMMUNICATING TECHNICAL DETAILS ABOUT CRYPTOGRAPHY SOFTWARE IS ILLEGAL IN SOME PARTS OF THE WORLD. SO, WHEN YOU IMPORT THIS PACKAGE TO YOUR COUNTRY, RE-DISTRIBUTE IT FROM THERE OR EVEN JUST EMAIL TECHNICAL SUGGESTIONS OR EVEN SOURCE PATCHES TO THE AUTHOR OR OTHER PEOPLE YOU ARE STRONGLY ADVISED TO PAY CLOSE ATTENTION TO ANY EXPORT/IMPORT AND/OR USE LAWS WHICH APPLY TO YOU. THE AUTHORS OF OPENSSL ARE NOT LIABLE FOR ANY VIOLATIONS YOU MAKE HERE. SO BE CAREFUL, RESPONSIBILITY IS YOURS.

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, at personal discretion).
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest OpenSSL version number is 0.9.6d

The procedures given in this chapter are likely to work on all Linux platforms, but we have only tested it on OpenNA Linux and Red Hat Linux.

Packages

The following is based on information listed by OpenSSL as of 2002/05/09. Please check http://www.openssl.org/ regularly for the latest status. We chose to install from source because it provides the facility to fine tune the installation.

Source code is available from:
OpenSSL Homepage: http://www.openssl.org/
OpenSSL FTP Site: 129.132.7.170
You must be sure to download: openssl-0.9.6d.tar.gz

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all the files installed onto the system if you want to update the package in the future. To solve this problem, it's a good idea to make a list of files on the system before you install OpenSSL, and then one afterwards, and then compare them using the **diff** utility to find out what files were placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep root]# find /* > OpenSSL1
  ```
- And the following one after you install the software:
  ```
  [root@deep root]# find /* > OpenSSL2
  ```
- Then use the following command to get a list of what changed:
  ```
  [root@deep root]# diff OpenSSL1 OpenSSL2 > OpenSSL-Installed
  ```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. In our example above, we use the /root directory of the system to store all the generated file lists.

Compiling - Optimizing & Installing OpenSSL

Below are the steps that you must make to configure, compile and optimize the OpenSSL software before installing it into your Linux system. First off, we install the program as user 'root' so as to avoid authorization problems.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- This can be done with the following commands:

```
[root@deep /]# cp openssl-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf openssl-version.tar.gz
```

Step 2
Next, move into the newly created OpenSSL source directory and perform the following steps to configure and optimize the software for your system.

- To move into the newly created OpenSSL directory use the following command:

```
[root@deep tmp]# cd openssl-0.9.6d/
```

Step 3
With OpenSSL, the optimization FLAGS should be changed in the "Configure" file of the program. It is in this file that we define the GCC optimizations we want to use related to the type of processor running in our system. OpenSSL is cryptography software and there are some optimization hacks that we can make that can significantly increase the performance of the program, therefore take the time to modify the "Configure" file of the software. This will be a benefit for you.

a) Edit the **Configure** file (vi +337 Configure) and change the following lines:

```
"linux-elf",    "gcc:-DL_ENDIAN -DTERMIO -O3 -fomit-frame-pointer -m486 -Wall::-
D_REENTRANT:-ldl:BN_LLONG ${x86_gcc_des}
${x86_gcc_opts}:${x86_elf_asm}:dlfcn:linux-shared:-
fPIC:.so.\$(SHLIB_MAJOR).\$(SHLIB_MINOR)",
```

To read:

```
"linux-elf",    "gcc:-DL_ENDIAN -DTERMIO -O3 -march=i686 -funroll-loops -fomit-
frame-pointer -Wall::-D_REENTRANT:-ldl:BN_LLONG ${x86_gcc_des}
${x86_gcc_opts}:${x86_elf_asm}:dlfcn:linux-shared:-
fPIC:.so.\$(SHLIB_MAJOR).\$(SHLIB_MINOR)",
```

b) Edit the **Configure** file (vi +338 Configure) and change the following lines:

```
"debug-linux-elf","gcc:-DBN_DEBUG -DREF_CHECK -DCONF_DEBUG -DBN_CTX_DEBUG -
DCRYPTO_MDEBUG -DL_ENDIAN -DTERMIO -g -m486 -Wall::-D_REENTRANT:-lefence -
ldl:BN_LLONG ${x86_gcc_des} ${x86_gcc_opts}:${x86_elf_asm}:dlfcn",
```

To read:

```
"debug-linux-elf","gcc:-DBN_DEBUG -DREF_CHECK -DCONF_DEBUG -DBN_CTX_DEBUG -
DCRYPTO_MDEBUG -DL_ENDIAN -DTERMIO -O3 -march=i686 -funroll-loops -fomit-frame-
pointer -Wall::-D_REENTRANT:-lefence -ldl:BN_LLONG ${x86_gcc_des}
${x86_gcc_opts}:${x86_elf_asm}:dlfcn",
```

c) Edit the **Configure** file (`vi +339 Configure`) and change the following lines:

```
"debug-linux-elf-noefence","gcc:-DBN_DEBUG -DREF_CHECK -DCONF_DEBUG -DBN_CTX_DEBUG
-DCRYPTO_MDEBUG -DL_ENDIAN -DTERMIO -g -m486 -Wall::-D_REENTRANT:-ldl:BN_LLONG
${x86_gcc_des} ${x86_gcc_opts}:${x86_elf_asm}:dlfcn",
```

To read:

```
"debug-linux-elf-noefence","gcc:-DBN_DEBUG -DREF_CHECK -DCONF_DEBUG -DBN_CTX_DEBUG
-DCRYPTO_MDEBUG -DL_ENDIAN -DTERMIO -O3 -march=i686 -funroll-loops -fomit-frame-
pointer -Wall::-D_REENTRANT:-ldl:BN_LLONG ${x86_gcc_des}
${x86_gcc_opts}:${x86_elf_asm}:dlfcn",
```

Step 4
By default, `OpenSSL` source files assume that our "`perl`" binary program is located under `/usr/local/bin/perl`. We must change this to reflect our environment variable.

- To point all `OpenSSL` script files to our "`perl`" binary, use the following command:
 `[root@deep openssl-0.9.6d]#` **`perl util/perlpath.pl /usr/bin/perl`**

Step 5
At this stage, it is time to configure and compile `OpenSSL` for our system.

- To configure and optimize `OpenSSL` use the following compilation lines:
  ```
  ./Configure linux-elf no-asm shared \
  --prefix=/usr \
  --openssldir=/usr/share/ssl
  ```

Step 6
Now, we must make a list of all files on the system before installing the software, and one afterwards, then compare them using the **diff** utility to find out what files are placed where and finally we install the `OpenSSL` software:

```
[root@deep openssl-0.9.6d]# LD_LIBRARY_PATH=`pwd` make all build-shared
[root@deep openssl-0.9.6d]# LD_LIBRARY_PATH=`pwd` make test apps tests
[root@deep openssl-0.9.6d]# cd
[root@deep root]# find /* > OpenSSL1
[root@deep root]# cd /var/tmp/openssl-0.9.6d/
[root@deep openssl-0.9.6d]# make install build-shared
[root@deep openssl-0.9.6d]# cd /usr/lib/
[root@deep lib]# mv libcrypto.so.0.9.6 ../../lib/
[root@deep lib]# mv libssl.so.0.9.6 ../../lib/
[root@deep lib]# ln -sf ../../lib/libcrypto.so.0.9.6 libcrypto.so
[root@deep lib]# ln -sf ../../lib/libcrypto.so.0.9.6 libcrypto.so.0
[root@deep lib]# ln -sf ../../lib/libssl.so.0.9.6 libssl.so
[root@deep lib]# ln -sf ../../lib/libssl.so.0.9.6 libssl.so.0
[root@deep lib]# mv /usr/share/ssl/man/man1/* /usr/share/man/man1/
[root@deep lib]# mv /usr/share/ssl/man/man3/* /usr/share/man/man3/
[root@deep lib]# mv /usr/share/ssl/man/man5/* /usr/share/man/man5/
[root@deep lib]# mv /usr/share/ssl/man/man7/* /usr/share/man/man7/
[root@deep lib]# rm -rf /usr/share/ssl/man/
[root@deep lib]# rm -rf /usr/share/ssl/lib/
[root@deep lib]# strip /usr/bin/openssl
[root@deep lib]# mkdir -p /usr/share/ssl/crl
[root@deep lib]# cd
[root@deep root]# find /* > OpenSSL2
[root@deep root]# diff OpenSSL1 OpenSSL2 > OpenSSL-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then test the OpenSSL libraries to finally install the binaries and any supporting files into the appropriate locations.

Step 7
Once the configuration, optimization, compilation, and installation of the OpenSSL software has completed, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete OpenSSL and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf openssl-version/
[root@deep tmp]# rm -f openssl-version.tar.gz
```

Configuring OpenSSL

After OpenSSL has been built and installed successfully on your system, your next step is to configure and customize the openssl.cnf and sign files to suit your needs.

- ✓ /usr/shared/ssl/openssl.cnf (The OpenSSL Configuration File)
- ✓ /usr/shared/ssl/misc/sign (A CA script file to sign certificates)

/usr/shared/ssl/openssl.cnf: The OpenSSL Configuration File

This is the general configuration file for OpenSSL, where you can configure the expiration date of your keys, the name of your organization, address and so on.

The most important parameters you may need to change will be in the **[CA_default]** and especially the **[req_distinguished_name]** sections of the file. We must change the default one to fit our requirements and operating system. The text in bold is the parts of the configuration file that must be customized and adjusted to satisfy our needs.

- Edit the **openssl.cnf** file (vi /usr/share/ssl/openssl.cnf) and set your needs.

```
#
# OpenSSL example configuration file.
# This is mostly being used for generation of certificate requests.
#

# This definition stops the following lines choking if HOME isn't
# defined.
HOME                    = .
RANDFILE                = $ENV::HOME/.rnd

# Extra OBJECT IDENTIFIER info:
#oid_file               = $ENV::HOME/.oid
oid_section             = new_oids

# To use this configuration file with the "-extfile" option of the
# "openssl x509" utility, name here the section containing the
# X.509v3 extensions to use:
# extensions            =
# (Alternatively, use a configuration file that has only
# X.509v3 extensions in its main [= default] section.)
```

```
[ new_oids ]

# We can add new OIDs in here for use by 'ca' and 'req'.
# Add a simple OID like this:
# testoid1=1.2.3.4
# Or use config file substitution like this:
# testoid2=${testoid1}.5.6

####################################################################
[ ca ]
default_ca      = CA_default            # The default ca section

####################################################################
[ CA_default ]

dir             = /usr/share/ssl        # Where everything is kept
certs           = $dir/certs            # Where the issued certs are kept
crl_dir         = $dir/crl              # Where the issued crl are kept
database        = $dir/ca.db.index      # database index file.
new_certs_dir   = $dir/ca.db.certs      # default place for new certs.

certificate     = $dir/certs/ca.crt     # The CA certificate
serial          = $dir/ca.db.serial     # The current serial number
crl             = $dir/crl.pem          # The current CRL
private_key     = $dir/private/ca.key   # The private key
RANDFILE        = $dir/ca.db.rand       # private random number file

x509_extensions = usr_cert              # The extentions to add to the cert

# Extensions to add to a CRL. Note: Netscape communicator chokes on V2 CRLs
# so this is commented out by default to leave a V1 CRL.
# crl_extensions        = crl_ext

default_days    = 365                   # how long to certify for
default_crl_days= 30                    # how long before next CRL
default_md      = md5                   # which md to use.
preserve        = no                    # keep passed DN ordering

# A few difference way of specifying how similar the request should look
# For type CA, the listed attributes must be the same, and the optional
# and supplied fields are just that :-)
policy          = policy_match

# For the CA policy
[ policy_match ]
countryName             = match
stateOrProvinceName     = match
organizationName        = match
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional

# For the 'anything' policy
# At this point in time, you must list all acceptable 'object'
# types.
[ policy_anything ]
countryName             = optional
stateOrProvinceName     = optional
localityName            = optional
organizationName        = optional
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional
```

```
#####################################################################
[ req ]
default_bits            = 1024
default_keyfile         = privkey.pem
distinguished_name      = req_distinguished_name
attributes              = req_attributes
x509_extensions = v3_ca # The extentions to add to the self signed cert

# Passwords for private keys if not present they will be prompted for
# input_password = secret
# output_password = secret

# This sets a mask for permitted string types. There are several options.
# default: PrintableString, T61String, BMPString.
# pkix   : PrintableString, BMPString.
# utf8only: only UTF8Strings.
# nombstr : PrintableString, T61String (no BMPStrings or UTF8Strings).
# MASK:XXXX a literal mask value.
# WARNING: current versions of Netscape crash on BMPStrings or UTF8Strings
# so use this option with caution!
string_mask = nombstr

# req_extensions = v3_req # The extensions to add to a certificate request

[ req_distinguished_name ]
countryName                     = Country Name (2 letter code)
countryName_default             = CA
countryName_min                 = 2
countryName_max                 = 2

stateOrProvinceName             = State or Province Name (full name)
stateOrProvinceName_default     = Quebec

localityName                    = Locality Name (eg, city)
localityName_default            = Montreal

0.organizationName              = Organization Name (eg, company)
0.organizationName_default      = OpenNA, Inc.

# we can do this but it is not needed normally :-)
#1.organizationName             = Second Organization Name (eg, company)
#1.organizationName_default     = World Wide Web Pty Ltd

organizationalUnitName          = Organizational Unit Name (eg, section)
organizationalUnitName_default  = Open Network Architecture

commonName                      = Common Name (eg, YOUR name)
commonName_default              = www.openna.com
commonName_max                  = 64

emailAddress                    = Email Address
emailAddress_default            = noc@openna.com
emailAddress_max                = 40

# SET-ex3                       = SET extension number 3

[ req_attributes ]
challengePassword               = A challenge password
challengePassword_min           = 8
challengePassword_max           = 20

unstructuredName                = An optional company name
```

```
[ usr_cert ]

# These extensions are added when 'ca' signs a request.

# This goes against PKIX guidelines but some CAs do it and some software
# requires this to avoid interpreting an end user certificate as a CA.

basicConstraints=CA:FALSE

# Here are some examples of the usage of nsCertType. If it is omitted
# the certificate can be used for anything *except* object signing.

# This is OK for an SSL server.
# nsCertType                    = server

# For an object signing certificate this would be used.
# nsCertType = objsign

# For normal client use this is typical
# nsCertType = client, email

# and for everything including object signing:
# nsCertType = client, email, objsign

# This is typical in keyUsage for a client certificate.
# keyUsage = nonRepudiation, digitalSignature, keyEncipherment

# This will be displayed in Netscape's comment listbox.
nsComment                       = "OpenSSL Generated Certificate"

# PKIX recommendations harmless if included in all certificates.
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer:always

# This stuff is for subjectAltName and issuerAltname.
# Import the email address.
# subjectAltName=email:copy

# Copy subject details
# issuerAltName=issuer:copy

#nsCaRevocationUrl              = http://www.domain.dom/ca-crl.pem
#nsBaseUrl
#nsRevocationUrl
#nsRenewalUrl
#nsCaPolicyUrl
#nsSslServerName

[ v3_req ]

# Extensions to add to a certificate request

basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment

[ v3_ca ]


# Extensions for a typical CA


# PKIX recommendation.
```

```
subjectKeyIdentifier=hash

authorityKeyIdentifier=keyid:always,issuer:always

# This is what PKIX recommends but some broken software chokes on critical
# extensions.
#basicConstraints = critical,CA:true
# So we do this instead.
basicConstraints = CA:true

# Key usage: this is typical for a CA certificate. However since it will
# prevent it being used as an test self-signed certificate it is best
# left out by default.
# keyUsage = cRLSign, keyCertSign

# Some might want this also
# nsCertType = sslCA, emailCA

# Include email address in subject alt name: another PKIX recommendation
# subjectAltName=email:copy
# Copy issuer details
# issuerAltName=issuer:copy

# DER hex encoding of an extension: beware experts only!
# obj=DER:02:03
# Where 'obj' is a standard or added object
# You can even override a supported extension:
# basicConstraints= critical, DER:30:03:01:01:FF

[ crl_ext ]

# CRL extensions.
# Only issuerAltName and authorityKeyIdentifier make any sense in a CRL.

# issuerAltName=issuer:copy
authorityKeyIdentifier=keyid:always,issuer:always
```

WARNING: You don't need to change all of the default options set in the file `openssl.cnf`; The configurations you usually change will be into the **`[ CA_default ]`** and **`[ req_distinguished_name ]`** sections of the file.

`/usr/share/ssl/misc/sign`: The `CA` Script File to Sign Certificates

`OpenSSL CA` command has some strange requirements and the default `OpenSSL` config doesn't allow one to easily use `OpenSSL CA` directly. It is for this reason that we don't use the files `CA.pl` or `CA.sh` to sign certificates.

Step 1
To solve the problem, we'll create and customize the `sign` script file below to replace them. Text in bold are the parts of the script that must be customized and adjusted to satisfy our needs.

- Create the **`sign`** script file (`touch /usr/share/ssl/misc/sign`) and add the following lines:

```
#!/bin/sh
##
```

```
##  sign.sh -- Sign a SSL Certificate Request (CSR)
##  Copyright (c) 1998-1999 Ralf S. Engelschall, All Rights Reserved.
##

#   argument line handling
CSR=$1
if [ $# -ne 1 ]; then
    echo "Usage: sign.sign <whatever>.csr"; exit 1
fi
if [ ! -f $CSR ]; then
    echo "CSR not found: $CSR"; exit 1
fi
case $CSR in
   *.csr ) CERT="`echo $CSR | sed -e 's/\.csr/.crt/'`" ;;
       * ) CERT="$CSR.crt" ;;
esac

#   make sure environment exists
if [ ! -d ca.db.certs ]; then
    mkdir ca.db.certs
fi
if [ ! -f ca.db.serial ]; then
    echo '01' >ca.db.serial
fi
if [ ! -f ca.db.index ]; then
    cp /dev/null ca.db.index
fi

#   create an own SSLeay config
cat >ca.config <<EOT
[ ca ]
default_ca    = CA_own
[ CA_own ]
dir           = /usr/share/ssl
certs         = /usr/share/ssl/certs
new_certs_dir = /usr/share/ssl/ca.db.certs
database      = /usr/share/ssl/ca.db.index
serial        = /usr/share/ssl/ca.db.serial
RANDFILE      = /usr/share/ssl/ca.db.rand
certificate   = /usr/share/ssl/certs/ca.crt
private_key   = /usr/share/ssl/private/ca.key
default_days = 365
default_crl_days               = 30
default_md    = md5
preserve      = no
policy        = policy_anything
[ policy_anything ]
countryName   = optional
stateOrProvinceName         = optional
localityName = optional
organizationName            = optional
organizationalUnitName      = optional
commonName    = supplied
emailAddress = optional
EOT

#  sign the certificate
echo "CA signing: $CSR -> $CERT:"
openssl ca -config ca.config -out $CERT -infiles $CSR
echo "CA verifying: $CERT <-> CA cert"
openssl verify -CAfile /usr/share/ssl/certs/ca.crt $CERT

#  cleanup after SSLeay
```

```
rm -f ca.config
rm -f ca.db.serial.old
rm -f ca.db.index.old

#  die gracefully
exit 0
```

Step 2
Once the script file has been created, it is important to make it executable and change its default permissions. Making this file executable will allow the system to run it, changing its default permission is to allow only the root user to change this file for security reason.

- To make this script executable and to change its default permissions, use the commands:
```
[root@deep /]# chmod 700 /usr/share/ssl/misc/sign
[root@deep /]# chown 0.0 /usr/share/ssl/misc/sign
```

OpenSSL Administrative Tools

Once your configuration options have been set in the openssl.cnf file, we can play with the OpenSSL utility. As an example, we'll show you how to create certificates for the Apache web server and your own CA (**C**ertifying **A**uthority) to sign your "**C**ertificate **S**igning **R**equest" yourself. All commands listed below are to be made in the **/usr/share/ssl** directory.

The Apache Key & CSR Generation:

The utility "openssl" that you use to generate RSA Private Keys (Key) and **C**ertificate **S**igning **R**equests (CSR) comes with OpenSSL and is usually installed under the directory /usr/bin on our Linux distribution. Below are the steps to create certificates for Apache.

Step 1
First you have to know the **F**ully **Q**ualified **D**omain **N**ame (FQDN) of the website/server for which you want to request a certificate. When you want to access your website/server through https://www.mydomain.com/ then the FQDN of your website is www.mydomain.com.

Step 2
Second, select five large and relatively random files from your hard drive (compressed log files are a good start) and put them under your /usr/share/ssl directory. These will act as your random seed enhancers. We refer to them as random1: random2:...: random5 below.

- To select five random files and put them under /usr/share/ssl, use the commands:
```
[root@deep /]# cp /var/log/boot.log /usr/share/ssl/random1
[root@deep /]# cp /var/log/cron /usr/share/ssl/random2
[root@deep /]# cp /var/log/dmesg /usr/share/ssl/random3
[root@deep /]# cp /var/log/messages /usr/share/ssl/random4
[root@deep /]# cp /var/log/secure /usr/share/ssl/random5
```

Step 3
Third, create the RSA private key protected with a pass-phrase for your web server. The command below will generate 1024 bit RSA Private Key and store it in www.mydomain.com.key.

It will ask you for a pass-phrase: use something secure and remember it. Your certificate will be useless without the key. If you don't want to protect your key with a pass-phrase (only if you absolutely trust that server machine, and you make sure the permissions are carefully set so only you can read that key) you can leave out the -des3 option below.

- To generate the Key, use the following command:

```
[root@deep /]# cd /usr/share/ssl/
[root@deep ssl]# openssl genrsa -des3 -rand
random1:random2:random3:random4:random5 -out www.mydomain.com.key 1024
123600 semi-random bytes loaded
Generating RSA private key, 1024 bit long modulus
.....................+++++
.....+++++
e is 65537 (0x10001)
Enter PEM pass phrase:
Verifying password - Enter PEM pass phrase:
```

WARNING: Please backup your **www.mydomain.com.key** file and remember the pass-phrase you had to enter at a secure location. A good choice is to backup this information onto a diskette or other removable media.

Step 4
Finally, generate a **C**ertificate **S**igning **R**equest (CSR) with the server RSA private key. The command below will prompt you for the X.509 attributes of your certificate. Remember to give the name "www.mydomain.com" when prompted for 'Common Name'. Do not enter your personal name here. We are requesting a certificate for a web server, so the 'Common Name' has to match the FQDN of your website (a requirement of the browsers).

- To generate the CSR, use the following command:

```
[root@deep ssl]# openssl req -new -key www.mydomain.com.key -out
www.mydomain.com.csr
Using configuration from /usr/share/ssl/openssl.cnf
Enter PEM pass phrase:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [CA]:
State or Province Name (full name) [Quebec]:
Locality Name (eg, city) [Montreal]:
Organization Name (eg, company) [OpenNA, Inc.]:
Organizational Unit Name (eg, section) [Open Network Architecture]:
Common Name (eg, YOUR name) [www.openna.com]:
Email Address [noc@openna.com]:

Please enter the following 'extra' attributes
```

```
to be sent with your certificate request
A challenge password []:.
An optional company name []:.
```

WARNING: Make sure you enter the FQDN (**F**ully **Q**ualified **D**omain **N**ame) of the server when OpenSSL prompts you for the "**Common Name**" (i.e. when you generate a CSR for a website which will be later accessed via https://www.mydomain.com/, enter "www.mydomain.com" here).

After the generation of your **C**ertificate **S**igning **R**equest (CSR), you must send this certificate to a commercial **C**ertifying **A**uthority (CA) like Thawte or Verisign for signing. You usually have to post the CSR into a web form, pay for the signing, await the signed certificate and store it into a "www.mydomain.com.crt" file. The result is then a real certificate, which can be used with Apache.

The CA Key & CRT Generation:

If you don't want to pay a commercial **C**ertifying **A**uthority (CA) to sign you certificates, you can use your own CA and now have to sign the CSR yourself by this CA. This solution is economical, and allows an organization to host their own CA server and generate as many certificates as they need for internal use without paying a cent to a commercial CA.

Unfortunately using your own CA to generate certificates causes problems in electronic commerce, because customers need to have some trust in your organization by the use of a recognized commercial CA. See below on how to sign a CSR with your CA yourself.

Step 1
As for the Apache web server above, the first step is to create the RSA private key protected with a pass-phrase for your CA. The command below will generate 1024 bit RSA Private Key and stores it in the file "ca.key". It will ask you for a pass-phrase: use something secure and remember it. Your certificate will be useless without the key.

- To create the RSA private key for your (CA), use the following command:

```
[root@deep /]# cd /usr/share/ssl/
[root@deep ssl]# openssl genrsa -des3 -out ca.key 1024
Generating RSA private key, 1024 bit long modulus
.........................+++++
.........................................+++++
e is 65537 (0x10001)
Enter PEM pass phrase:
Verifying password - Enter PEM pass phrase:
```

WARNING: Please backup your "**ca.key**" file and remember the pass-phrase you had to enter at a secure location. A good choice is to backup this information onto a diskette or other removable media.

Step 2
Now, we must create a self-signed (CA) certificate (x509 structure) with the RSA key of the CA. The req command creates a self-signed certificate when the -x509 switch is used.

- To create a self-signed (CA) certificate, use the following command:

```
[root@deep ssl]# openssl req -new -x509 -days 365 -key ca.key -out ca.crt
Using configuration from /usr/share/ssl/openssl.cnf
Enter PEM pass phrase:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [CA]:
State or Province Name (full name) [Quebec]:
Locality Name (eg, city) [Montreal]:
Organization Name (eg, company) [OpenNA, Inc.]:
Organizational Unit Name (eg, section) [Open Network]:Sales Dept
Common Name (eg, YOUR name) [www.openna.com]:
Email Address [noc@openna.com]:sales@openna.com
```

Step 3
Once the self-signed (CA) certificate has been created, we must place all certificates and CA files into their appropriate directories.

- To place the files into their appropriate directories, use the following commands:

```
[root@deep ssl]# mv www.mydomain.com.key private/
[root@deep ssl]# mv ca.key private/
[root@deep ssl]# mv ca.crt certs/
```

Step 4
Finally, you can use this CA to sign all the servers CSR's in order to create real SSL Certificates for use inside the web server (assuming you already have a www.mydomain.com.csr at hand). We must also prepare the script "**sign**" for signing.

- To sign server CSR's in order to create real SSL Certificates, use the following command:

```
[root@deep ssl]# /usr/share/ssl/misc/sign www.mydomain.com.csr
CA signing: www.mydomain.com.csr -> www.mydomain.com.crt:
Using configuration from ca.config
Enter PEM pass phrase:
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName               :PRINTABLE:'CA'
stateOrProvinceName       :PRINTABLE:'Quebec'
localityName              :PRINTABLE:'Montreal'
organizationName          :PRINTABLE:'OpenNA, Inc.'
organizationalUnitName    :PRINTABLE:'Open Network Architecture'
commonName                :PRINTABLE:'www.openna.com'
emailAddress              :IA5STRING:'noc@openna.com'
Certificate is to be certified until Oct  18 14:59:29 2001 GMT (365 days)
Sign the certificate? [y/n]:y
```

```
1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
CA verifying: www.mydomain.com.crt <-> CA cert
www.mydomain.com.crt: OK
```

This signs the CSR and results in a "www.mydomain.com.crt" file. Move this file to its appropriate directory as follows.

- To move the CRT file to its appropriate directory, use the following command:
 `[root@deep ssl]# mv www.mydomain.com.crt certs/`

Now you have two files: "**www.mydomain.com.key**" and "**www.mydomain.com.crt**". These can now, for example, be used as follows, inside the virtual host section of your Apache server's httpd.conf file:

```
SSLCertificateFile      /usr/share/ssl/certs/www.mydomain.com.crt
SSLCertificateKeyFile   /usr/share/ssl/private/www.mydomain.com.key
```

In this example, www.mydomain.com.crt is our web server **C**ertificate **S**igning **R**equest Public Key, and www.mydomain.com.key is our web server RSA Private Key.

The **www.mydomain.com.csr** file is no longer needed; we can remove it from the system.

- To remove this file from the system, use the following command:
 `[root@deep ssl]# rm -f www.mydomain.com.csr`

WARNING: If you receive an error message during the signing of the certificate, it's probably because you've entered the wrong FQDN (Fully Qualified Domain Name) for the server when OpenSSL prompted you for the "**Common Name**"; the "**Common Name**" must be something like "www.mydomain.com" and not "mydomain.com". Also, since you generate both the certificate and the CA certificate, it's important that at least ONE piece of information differs between both files, or you may encounter problems during the signature of the certificate request.

Securing OpenSSL

This small section deals specifically with actions we can take to improve and tighten security under OpenSSL. It's important to note that we refer to the features available within the base installed program and not to any additional software.

Changing the default mode of OpenSSL keys:

Make your keys "Read and Write" only by the super-user "root". This is important because no one needs to touch these files.

- To make your keys "read and Write" only by "root", use the following commands:

```
[root@deep /]# chmod 750 /usr/share/ssl/private/
[root@deep /]# chmod 400 /usr/share/ssl/certs/ca.crt
[root@deep /]# chmod 400 /usr/share/ssl/certs/www.mydomain.com.crt
[root@deep /]# chmod 400 /usr/share/ssl/private/ca.key
[root@deep /]# chmod 400 /usr/share/ssl/private/www.mydomain.com.key
```

Some possible uses of OpenSSL software

OpenSSL can be used to:

1. Creation of your own Certifying Authority Server.
2. Creation of RSA, DH and DSA key parameters.
3. Creation of X.509 certificates, CSRs and CRLs.
4. Calculation of Message Digest.
5. Encryption and Descryptiion with Ciphers.
6. SSL/TLS Client and Server Tests.
7. Handling of S/MIME signed or encrypted mail.
8. Provide data confidentiality, integrity, authentication, and electronic signature in transmission for the users.
9. Secure electronic commerce transactions.

CHAPTER 16

OpenSSH

IN THIS CHAPTER

1. **Compiling - Optimizing & Installing `OpenSSH`**
2. **Configuring `OpenSSH`**
3. **Running `OpenSSH` in a chroot jail**
4. **Creating `OpenSSH` private & public keys**
5. **`OpenSSH` Users Tools**

Linux OpenSSH

Abstract
As illustrated in the chapter related to the Linux installation, many network services including, but not limited to, telnet, rsh, rlogin, or rexec are vulnerable to electronic eavesdropping. As a consequence, anyone who has access to any machine connected to the network can listen in on its network communications and get your password, as well as any other private information that is sent over the network in plain text.

Currently the Telnet program is indispensable for daily administration tasks, but it is insecure since it transmits your password in plain text over the network and allows any listener to capture your password and then use your account to do anything he likes. To solve this problem we must find either another way, or another program, to replace it. Fortunately OpenSSH is a truly seamless and secure replacement of old, insecure and obsoletes remote login programs such as telnet, rlogin, rsh, rdist, or rcp.

SSH (**S**ecure **Sh**ell) is a program to log into another computer over a network, to execute commands on a remote machine, and to move files from one machine to another. It provides strong authentication and secure communications over insecure channels. It is intended as a replacement for rlogin, rsh, rcp, and rdist.

These installation instructions assume
Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, at personal discretion).
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest OpenSSH version number is 3.4p1

The procedures given in this chapter are likely to work on all Linux platforms, but we have only tested it on OpenNA Linux and Red Hat Linux.

Packages
The following is based on information listed by OpenSSH as of 2002/06/26. Please check http://www.openssh.com/ regularly for the latest status. We chose to install from source because it provides the facility to fine tune the installation.

Source code is available from:
OpenSSH Homepage: http://www.openssh.com/
OpenSSH FTP Site: 129.128.5.191
You must be sure to download: openssh-3.4p1.tar.gz

> **NOTE:** Don't forget to download the portable version (the **p** suffix) of OpenSSH tarball for Linux. There is strictly OpenBSD-based development of this software and another one known as portable version, which runs on many operating systems (these are known as the **p** releases, and named like "OpenSSH 3.4p1").

Prerequisites

OpenSSH requires that the listed software below be already installed on your system to be able to compile successfully. If this is not the case, you must install it from your Linux CD-ROM or source archive files. Please make sure you have this program installed on your machine before you proceed with this chapter.

- ✓ OpenSSL is required to run OpenSSH on your system.

NOTE: For more information on OpenSSL software, see its related chapter in this book. Even if you don't need to use OpenSSL software to create or hold encrypted key files, it's important to note that OpenSSH requires its libraries files to be able to work.

Pristine source

As we don't use the RPM package to install this program, it will be difficult for you to locate all the files installed onto the system if you want to update the package in the future. To solve this problem, it's a good idea to make a list of files on the system before you install OpenSSH, and then one afterwards, and then compare them using the **diff** utility to find out what files were placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep root]# find /* > OpenSSH1
  ```
- And the following one after you install the software:
  ```
  [root@deep root]# find /* > OpenSSH2
  ```
- Then use the following command to get a list of what changed:
  ```
  [root@deep root]# diff OpenSSH1 OpenSSH2 > OpenSSH-Installed
  ```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. In our example above, we use the /root directory of the system to store all the generated file lists.

Compiling - Optimizing & Installing OpenSSH

Below are the steps that you must make to configure, compile and optimize the OpenSSH server software before installing it on your system. First off, we install the program as user 'root' so as to avoid any authorization problems.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- This can be done with the following commands:
  ```
  [root@deep /]# cp openssh-version.tar.gz /var/tmp/
  [root@deep /]# cd /var/tmp/
  [root@deep tmp]# tar xzpf openssh-version.tar.gz
  ```

Step 2
In order to check that the version of OpenSSH, which you are, going to install, is an original and unmodified one, please check the supplied signature with the GPG key of OpenSSH available on the OpenSSH website.

To get a GPG key copy of OpenSSH, please point your browser to the following URL: ftp://ftp.openbsd.org/pub/OpenBSD/OpenSSH/portable/openssh-3.4p1.tar.gz.sig. For more information about how to use this key for verification, see the GnuPG chapter in this book.

Step 3
OpenSSH needs a UID and GID to properly run on the system but this UID/GID cannot run as super-user root; for this reason we must create a special user with no shell privileges on the system for running sshd daemon. This is required by the privilege separation feature of OpenSSH by which operations that require root privilege are performed by a separate privileged monitor process.

- To create this special OpenSSH user on OpenNA Linux, use the following command:

```
[root@deep tmp]# groupadd -g 39 sshd > /dev/null 2>&1 || :
[root@deep tmp]# useradd -c "SSH Server" -d /var/empty -g 39 -s
/bin/false -u 39 sshd > /dev/null 2>&1 || :
```

- To create this special OpenSSH user on Red Hat Linux, use the following command:

```
[root@deep tmp]# groupadd -g 39 sshd > /dev/null 2>&1 || :
[root@deep tmp]# useradd -u 39 -g 39 -s /bin/false -M -r -d /var/empty
sshd > /dev/null 2>&1 || :
```

The above command will create a null account, with no password, no valid shell, no files owned-nothing but a UID and a GID for the program. Remember that OpenSSH daemon does not need to have a shell account on the server.

Step 4
Now, edit the **shells** file (vi /etc/shells) and add a non-existent shell name "/bin/false", which is the one we used in the useradd command above.

```
[root@deep tmp]# vi /etc/shells
/bin/bash2
/bin/bash
/bin/sh
/bin/false ← This is our added no-existent shell
```

Patching OpenSSH to run in chroot jail mode for some users:

There is an external patch available for OpenSSH that allow us to compile OpenSSH with chroot jail support. If you are interested in compiling OpenSSH to run in chroot jail environment for some of your users, then I recommend you to follow these steps. If you don't want to compile OpenSSH with chroot jail support, you can simply skip these steps and go directly to next section where we will compile the software for our system.

For OpenSSH to run and work in chroot jail mode, you have to be sure that you have recompiled your Linux kernel without the Grsecurity option that allows us to enable chroot jail restrictions protection on the system. You should be sure that "Chroot jail restrictions (CONFIG_GRKERNSEC_CHROOT) [N/y/?]" is NOT enable or nothing will work.

Step 1
First of, we have to retrieve the OpenSSH chroot patch available on the Internet. This patch can be downloaded from the following location: http://chrootssh.sourceforge.net/

Step 2
Once you have a copy of this patch, you should move it under the /var/tmp directory and patch your OpenSSH source files.

- This can be done with the following commands:

```
[root@deep /]# mv osshChroot-3.4.diff /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# patch -p0 < osshChroot-3.4.diff
```

NOTE: It's important to note that the version number of the OpenSSH chroot patch that you have to download from the Internet must match the version number of the OpenSSH software you intended to install. For example, if the version number of OpenSSH is 3.4p1, you should download the newer OpenSSH chroot patch that matches this number.

Step 3
After that, move into the newly created OpenSSH source directory and perform the following steps to configure and optimize the software for your system.

- To move into the newly created OpenSSH directory use the following command:

```
[root@deep tmp]# cd openssh-3.4p1/
```

- To configure and optimize OpenSSH use the following compilation lines:

```
CFLAGS="-O2 -march=i686 -funroll-loops"; export CFLAGS
./configure \
--prefix=/usr \
--sysconfdir=/etc/ssh \
--libexecdir=/usr/libexec/openssh \
--mandir=/usr/share/man \
--with-pam \
--with-ipaddr-display \
--with-ipv4-default \
--with-md5-passwords \
--with-zlib
```

This tells OpenSSH to set itself up for this particular configuration setup with:

- Enable PAM support.
- Use the IP address instead of hostname.
- Use IPv4 by connections.
- Enable use of MD5 passwords.
- Use zlib for transport compression.

NOTE: Pay special attention to the compile CFLAGS line above. We optimize OpenSSH for an i686 CPU architecture with the parameter "-march=i686". Please don't forget to adjust this CFLAGS line to reflect your own system and architecture.

Step 4
Now, we must make a list of all existing files on the system before installing the software and one afterwards then compare them using the **diff** utility to find out what files are placed where and finally install the OpenSSH Server:

```
[root@deep openssh-3.4p1]# make
[root@deep openssh-3.4p1]# cd
[root@deep root]# find /* > OpenSSH1
[root@deep root]# cd /var/tmp/openssh-3.4p1/
[root@deep openssh-3.4p1]# make install
[root@deep openssh-3.4p1]# mkdir /var/empty
[root@deep openssh-3.4p1]# chown root.sys /var/empty
[root@deep openssh-3.4p1]# cd
[root@deep root]# find /* > OpenSSH2
[root@deep root]# diff OpenSSH1 OpenSSH2 > OpenSSH-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

Step 5
Once the configuration, optimization, compilation, and installation of the OpenSSH software has been accomplished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete OpenSSH and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf openssh-version/
[root@deep tmp]# rm -f openssh-version.tar.gz
```

The rm command as used above will remove all the source files we have used to compile and install OpenSSH. It will also remove the OpenSSH compressed archive from the /var/tmp directory.

Configuring OpenSSH

After OpenSSH has been built and installed successfully in your system, your next step is to configure and customize its configuration files to fit your needs.

- ✓ /etc/ssh/sshd_config: (The OpenSSH Server Configuration File)
- ✓ /etc/ssh/ssh_config: (The OpenSSH Client Configuration File)
- ✓ /etc/pam.d/sshd: (The OpenSSH PAM Support Configuration File)
- ✓ /etc/init.d/sshd: (The OpenSSH Initialization File)

/etc/ssh/sshd_config: The OpenSSH Server Configuration File

The sshd_config file is the system-wide server configuration file for OpenSSH which allows you to set options that modify the operation of the sshd daemon. This file contains keyword-value pairs, one per line, with keywords being case insensitive.

Here are the most important keywords to configure your sshd server for maximum security; a complete listing and/or special requirements are available in the manual page for sshd (8). We must change the default one to fit our requirements and operating system. The text in bold are the parts of the configuration file that must be customized and adjusted to satisfy our needs.

- Edit the **sshd_config** file (`vi /etc/ssh/sshd_config`). Below is what we recommend you enter:

```
# This is ssh server systemwide configuration file.

Port                            22
Protocol                        2,1
ListenAddress                   207.35.78.3
HostKey                         /etc/ssh/ssh_host_key
HostKey                         /etc/ssh/ssh_host_rsa_key
HostKey                         /etc/ssh/ssh_host_dsa_key
ServerKeyBits                   768
LoginGraceTime                  60
KeyRegenerationInterval         3600
PermitRootLogin                 no
IgnoreRhosts                    yes
IgnoreUserKnownHosts            yes
StrictModes                     yes
X11Forwarding                   no
X11DisplayOffset                10
PrintMotd                       yes
KeepAlive                       yes
SyslogFacility                  AUTHPRIV
LogLevel                        INFO
RhostsAuthentication            no
RhostsRSAAuthentication         no
RSAAuthentication               yes
PasswordAuthentication          no
PermitEmptyPasswords            no
AllowUsers                      sysadmin
UsePrivilegeSeparation          yes
Subsystem                       sftp /usr/libexec/openssh/sftp-server
```

This tells the `sshd_config` file to set itself up for this particular configuration with:

`Port 22`
The option "`Port`" specifies on which port number the `sshd` daemon listens for incoming connections. The default port is 22.

`Protocol 2,1`
This option "`Protocol`" specifies the protocol versions `sshd` should support in order of preference. In our configuration the default is "`2,1`". This means that `sshd` tries version 2 and falls back to version 1 if version 2 is not available. Depending of the `ssh` client version you use to connect, you may need to invert this order but you can connect with `ssh` client version 1 even if the order is "`2,1`".

`ListenAddress 207.35.78.3`
The option "`ListenAddress`" specifies the `IP` address of the interface network on which the `sshd` daemon server socket is bound. The default is "`0.0.0.0`"; to improve security you may specify only the required ones to limit possible IP addresses. This is a security feature.

```
HostKey /etc/ssh/ssh_host_key
HostKey /etc/ssh/ssh_host_dsa_key
HostKey /etc/ssh/ssh_host_rsa_key
```
These options specify the location containing the different private host keys. If you have compiled `OpenSSH` as described in this book, then the default ones are correct.

`ServerKeyBits 768`
The option "`ServerKeyBits`" specifies how many bits to use in the server key. These bits are used when the daemon starts to generate its `RSA` key.

`LoginGraceTime 60`
The option "`LoginGraceTime`" specifies how long in seconds after a connection request the server will wait before disconnecting, if the user has not successfully logged in. A low value is recommended for this setting. Imagine what 1024 simulated connections at the same time can do to the other processes on your server.

`KeyRegenerationInterval 3600`
The option "`KeyRegenerationInterval`" specifies how long in seconds the server should wait before automatically regenerated its key. This is a security feature to prevent decrypting captured sessions.

`PermitRootLogin no`
The option "`PermitRootLogin`" specifies whether super-user "`root`" can log in using `ssh`. Never say "`yes`" to this option. It is safer to log in with a regular `UID` and then `su` or `sudo` to super-user "`root`". This is a security feature.

`IgnoreRhosts yes`
The option "`IgnoreRhosts`" specifies whether the `rhosts` or `shosts` files should not be used in authentication. For security reasons it is recommended to NOT use `rhosts` or `shosts` files for authentication. This is a security feature.

`IgnoreUserKnownHosts yes`
The option "`IgnoreUserKnownHosts`" specifies whether the `sshd` daemon should ignore the user's `$HOME/.ssh/known_hosts` file during `RhostsRSAAuthentication`. Since we don't allow `.rhosts` files on our server, it is safe to say "`yes`" here. This is a security feature.

`StrictModes yes`
The option "`StrictModes`" specifies whether `sshd` should check user's permissions in their home directory and `rhosts` files before accepting login. This option must always be set to "`yes`" because sometimes users may accidentally leave their directory or files world-writable. This is a security feature.

`X11Forwarding no`
The option "`X11Forwarding`" specifies whether `X11` forwarding should be enabled or not on this server. Since we setup a server without a `GUI` installed on it, we can safely turn this option off.

`PrintMotd yes`
The option "`PrintMotd`" specifies whether the `sshd` daemon should print the contents of the `/etc/motd` file when a user logs in interactively. The `/etc/motd` file is also known as "the message of the day".

`SyslogFacility AUTHPRIV`
The option "`SyslogFacility`" specifies the facility code used when logging messages from `sshd`. The facility specifies the subsystem that produced the message--in our case, `AUTHPRIV`.

`LogLevel INFO`
The option "`LogLevel`" specifies the level that is used when logging messages from `sshd`. `INFO` is a good choice. See the manual page for `sshd` for more information on other possibilities.

`RhostsAuthentication no`
The option "`RhostsAuthentication`" specifies whether `sshd` can try to use `rhosts` based authentication. Because `rhosts` authentication is insecure you shouldn't use this option. This is a security feature.

`RhostsRSAAuthentication no`
The option "`RhostsRSAAuthentication`" specifies whether to try `rhosts` authentication in concert with `RSA` host authentication. This is a security feature.

`RSAAuthentication yes`
The option "`RSAAuthentication`" specifies whether to try `RSA` authentication. It is important to note that it is reserved for the `SSH1` protocol only. This option must be set to "`yes`" for enhanced security in your sessions if you use `SSH1` and only `SSH1`, since it doesn't apply for the `SSH2` protocol (`SSH2` use `DSA` instead of `RSA`). `RSA` uses public and private key pairs created with the `ssh-keygen` utility for authentication purposes.

`PasswordAuthentication no`
The option "`PasswordAuthentication`" specifies whether we should use password-based authentication. For strong security, this option must always be set to "`no`". You should put '`PasswordAuthentication no`' in the `sshd_config` file, otherwise people might try to guess the password for the user. With '`PasswordAuthentication no`', your public key must be on the computer or no login is allowed: that's what we want. This is a security feature.

`PermitEmptyPasswords no`
This option "`PermitEmptyPasswords`" is closely related with the above option "`PasswordAuthentication`" and specifies whether, if password authentication is allowed, the server should allow logging in to accounts with a null password. Since we do not allow password authentication in the server, we can safety turn off this option. This is a security feature.

`AllowUsers sysadmin`
This option "`AllowUsers`" specifies and controls which users can access `ssh` services. Multiple users can be specified, separated by spaces. This is a security feature.

`UsePrivilegeSeparation yes`
This option "`UsePrivilegeSeparation`" is used to contain and restrict the effects of programming errors. A bug in the unprivileged child process does not result in a system compromise. Previously any corruption in the `sshd` daemon could lead to an immediate remote `root` compromise if it happened before authentication and to local `root` compromise if it happened after authentication. The "Privilege Separation" feature of `OpenSSH` will make such compromise very difficult if not impossible. This is a security feature.

/etc/ssh/ssh_config: The OpenSSH Client Configuration File

The ssh_config file is the system-wide client configuration file for OpenSSH which allows you to set options that modify the operation of the SSH client programs. The file contains keyword-value pairs, one per line, with keywords being case insensitive.

Here are the most important keywords to configure your ssh client for maximum security; a complete listing and/or special requirements is available in the manual page for ssh (1). We must change the default ones to fit our requirements and operating system. The text in bold is the parts of the configuration file that must be customized and adjusted to satisfy your needs.

- Edit the **ssh_config** file (vi /etc/ssh/ssh_config) and set your needs. Below is what we recommend you enter:

```
# Site-wide defaults for various options

Host                            *
ForwardAgent                    no
ForwardX11                      no
RhostsAuthentication            no
RhostsRSAAuthentication         no
RSAAuthentication               yes
PasswordAuthentication          no
FallBackToRsh                   no
UseRsh                          no
BatchMode                       no
CheckHostIP                     yes
StrictHostKeyChecking           yes
IdentityFile                    ~/.ssh/identity
IdentityFile                    ~/.ssh/id_dsa
IdentityFile                    ~/.ssh/id_rsa
Port                            22
Protocol                        2,1
Cipher                          blowfish
EscapeChar                      ~
```

This tells the ssh_config file to set itself up for this particular configuration with:

Host *
This option "Host" restricts all forwarded declarations and options in the configuration file to be only for those hosts that match one of the patterns given after the keyword. The pattern "*" means for all hosts up to the next "Host" keyword. With this option you can set different declarations for different hosts in the same ssh_config file. In particular, I find it useful when you want to automate backups over the network with SSH and don't want to supply the users password. In this way we can build a new section reserved for this and disable functions that ask for passwords for the specified host in question.

ForwardAgent no
This option "ForwardAgent" specifies which connection authentication agent (if any) should be forwarded to the remote machine.

ForwardX11 no
This option "ForwardX11" is for people that use the Xwindow GUI and want to automatically redirect X11 sessions to the remote machine. Since we have a server and it doesn't have GUI installed on it, we can safely turn this option off.

RhostsAuthentication no
This option "RhostsAuthentication" specifies whether we can try to use rhosts based authentication. Because rhosts authentication is insecure you shouldn't use this option. This is a security feature.

RhostsRSAAuthentication no
This option "RhostsRSAAuthentication" specifies whether or not to try rhosts authentication in concert with RSA host authentication. Evidently our answer is "no". This is a security feature.

RSAAuthentication yes
This option "RSAAuthentication" specifies whether to try RSA authentication. It is important to note that it is reserved for the SSH1 protocol only. This option must be set to yes for better security in your sessions if you use SSH1 and only SSH1 since it doesn't applies for SSH2 protocol (SSH2 use DSA instead of RSA). RSA use public and private key pairs created with the ssh-keygen utility for authentication purposes. Enable only if you connect to OpenSSH with client software that use SSH1 protocol.

PasswordAuthentication no
This option "PasswordAuthentication" specifies whether we should use password-based authentication. For strong security, this option must always be set to no. You should put 'PasswordAuthentication no' in the sshd_config file, otherwise people might try to guess the password for the user. With 'PasswordAuthentication no', your public key must be on the computer or no login is allowed: that's what we want. This is a security feature.

FallBackToRsh no
This option "FallBackToRsh" specifies that if a connection with ssh daemon fails rsh should automatically be used instead. Recalling that rsh service is insecure, this option must always be set to "no". This is a security feature.

UseRsh no
This option "UseRsh" specifies that rlogin/rsh services should be used on this host. As with the FallBackToRsh option, it must be set to "no" for obvious reasons. This is a security feature.

BatchMode no
This option "BatchMode" specifies whether a username and password querying on connect will be disabled. This option is useful when you create scripts and don't want to supply the password. (e.g. Scripts that use the scp command to make backups over the network).

CheckHostIP yes
This option "CheckHostIP" specifies whether or not ssh will additionally check the host IP address that connect to the server to detect DNS spoofing. It's recommended that you set this option to "yes" but on the other hand you can lose some performance doing this.

StrictHostKeyChecking yes
This option "StrictHostKeyChecking" specifies whether or not ssh will automatically add new host keys to the $HOME/.ssh/known_hosts file. This option, when set to "yes", provides the maximum protection against Trojan horse attacks. One interesting procedure with this option is to set it to "no" at the beginning, allow ssh to add automatically all common hosts to the host file as they are connected to, and then return to set it to "yes" to take advantage of its feature. This is a security feature.

```
IdentityFile ~/.ssh/identity
IdentityFile ~/.ssh/id_dsa
IdentityFile ~/.ssh/id_rsa
```
These options specify alternate multiple authentication identity files to read.

```
Port 22
```
This option "Port" specifies on which port number ssh connects to on the remote host. The default port is 22.

```
Protocol 2,1
```
This option "Protocol" specifies the protocol versions ssh should support in order of preference. In our configuration the default is "2,1". This means that ssh tries version 2 and falls back to version 1 if version 2 is not available. Depending of the ssh client version you use to connect, you may need to invert this order but you can connect with ssh client version 1 even if the order is "2,1".

```
Cipher blowfish
```
This option "Cipher" specifies what cipher should be used for encrypting sessions. The blowfish use 64-bit blocks and keys of up to 448 bits.

```
EscapeChar ~
```
This option "EscapeChar" specifies the session escape character for suspension.

/etc/pam.d/sshd: The OpenSSH PAM Support Configuration File

For increased security of OpenSSH, we have compiled it to use the PAM mechanism for password authentication.

Step 1
To be able to use this feature, we must create the /etc/pam.d/sshd file and add the following parameters inside it.

- Create the **sshd** file (touch /etc/pam.d/sshd) and add the following lines:

```
#%PAM-1.0
auth       required     /lib/security/pam_stack.so service=system-auth
auth       required     /lib/security/pam_nologin.so
account    required     /lib/security/pam_stack.so service=system-auth
account    required     /lib/security/pam_access.so
account    required     /lib/security/pam_time.so
password   required     /lib/security/pam_stack.so service=system-auth
session    required     /lib/security/pam_stack.so service=system-auth
session    required     /lib/security/pam_limits.so
```

Step2
Now, set the permissions of the sshd file to be (0640/-rw-r-----) and owned by the super-user 'root' for security reasons.

- To change the permissions and ownership of the sshd file, use the commands:

```
[root@deep /]# chmod 640 /etc/pam.d/sshd
[root@deep /]# chown 0.0 /etc/pam.d/sshd
```

/etc/init.d/sshd: The OpenSSH Initialization File

The /etc/init.d/sshd script file is responsible to automatically starting and stopping the OpenSSH server on your Linux system. Loading the sshd daemon as a standalone daemon will eliminate load time and will even reduce swapping since non-library code will be shared.

Please note that the following script is suitable for Linux operating systems that use SystemV. If you Linux system use some other methods like BSD, you'll have to adjust the script bellow to make it work for you.

Step 1
Create the **sshd** script file (touch /etc/init.d/sshd) and add the following lines:

```
#!/bin/bash

# This shell script takes care of starting and stopping OpenSSH.
#
# chkconfig: 2345 55 25
# description: OpenSSH is a program thas allows to establish a secure remote \
#              connection to a server.
#
# processname: sshd
# config: /etc/ssh/ssh_host_key
# config: /etc/ssh/ssh_host_key.pub
# config: /etc/ssh/ssh_random_seed
# config: /etc/ssh/sshd_config
# pidfile: /var/run/sshd.pid

# Source function library.
. /etc/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Source OpenSSH configureation.
if [ -f /etc/sysconfig/sshd ] ; then
        . /etc/sysconfig/sshd
fi

RETVAL=0

# Some functions to make the below more readable.
KEYGEN=/usr/bin/ssh-keygen
RSA1_KEY=/etc/ssh/ssh_host_key
RSA_KEY=/etc/ssh/ssh_host_rsa_key
DSA_KEY=/etc/ssh/ssh_host_dsa_key
PID_FILE=/var/run/sshd.pid
my_success() {
  local msg
  if [ $# -gt 1 ]; then
    msg="$2"
  else
    msg="done"
  fi
  case "`type -type success`" in
    function)
      success "$1"
    ;;
    *)
      echo -n "${msg}"
    ;;
  esac
```

```
}
my_failure() {
  local msg
  if [ $# -gt 1 ]; then
    msg="$2"
  else
    msg="FAILED"
  fi
  case "`type -type failure`" in
    function)
      failure "$1"
    ;;
    *)
      echo -n "${msg}"
    ;;
  esac
}
do_rsa1_keygen() {
        if ! test -f $RSA1_KEY ; then
                echo -n "Generating SSH1 RSA host key: "
                if $KEYGEN -q -t rsa1 -f $RSA1_KEY -C '' -N '' >&/dev/null;
then
                        my_success "RSA1 key generation"
                        echo
                else
                        my_failure "RSA1 key generation"
                        echo
                        exit 1
                fi
        fi
}
do_rsa_keygen() {
        if ! test -f $RSA_KEY ; then
                echo -n "Generating SSH2 RSA host key: "
                if $KEYGEN -q -t rsa -f $RSA_KEY -C '' -N '' >&/dev/null; then
                        my_success "RSA key generation"
                        echo
                else
                        my_failure "RSA key generation"
                        echo
                        exit 1
                fi
        fi
}
do_dsa_keygen() {
        if ! test -f $DSA_KEY ; then
                echo -n "Generating SSH2 DSA host key: "
                if $KEYGEN -q -t dsa -f $DSA_KEY -C '' -N '' >&/dev/null; then
                        my_success "DSA key generation"
                        echo
                else
                        my_failure "DSA key generation"
                        echo
                        exit 1
                fi
        fi
}
do_restart_sanity_check() {
        sshd -t
        RETVAL=$?
        if [ ! "$RETVAL" = 0 ]; then
                my_failure "Configuration file or keys"
                echo
```

```
                exit $RETVAL
        fi
}


case "$1" in
        start)
                # Create keys if necessary
                do_rsa1_keygen;
                do_rsa_keygen;
                do_dsa_keygen;

                echo -n "Starting SSHD: "
                if [ ! -f $PID_FILE ] ; then
                        sshd $OPTIONS
                        RETVAL=$?
                        if [ "$RETVAL" = "0" ] ; then
                                my_success "sshd startup" "sshd"
                                touch /var/lock/subsys/sshd
                        else
                                my_failure "sshd startup" ""
                        fi
                fi
                echo
                ;;
        stop)
                echo -n "Shutting down SSHD: "
                if [ -f $PID_FILE ] ; then
                        killproc sshd
                        RETVAL=$?
                        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/sshd
                fi
                echo
                ;;
        restart)
                do_restart_sanity_check
                $0 stop
                $0 start
                RETVAL=$?
                ;;
        condrestart)
                if [ -f /var/lock/subsys/sshd ] ; then
                        do_restart_sanity_check
                        $0 stop
                        $0 start
                        RETVAL=$?
                fi
                ;;
        status)
                status sshd
                RETVAL=$?
                ;;
        *)
                echo "Usage: sshd {start|stop|restart|status|condrestart}"
                exit 1
                ;;
esac
exit $RETVAL
```

Step 2
Once the /etc/init.d/sshd script file has been created, it is important to make it executable, change its default permissions, create the necessary links and start it. Making this file executable will allow the system to run it, changing its default permission is to allow only the root user to change this file for security reason, and creation of the symbolic links will let the process control initialization of Linux to start the program automatically for you at each system boot.

- To make this script executable and to change its default permissions, use the commands:
```
[root@deep /]# chmod 700 /etc/init.d/sshd
[root@deep /]# chown 0.0 /etc/init.d/sshd
```

- To create the symbolic rc.d links for OpenSSH, use the following commands:
```
[root@deep /]# chkconfig --add sshd
[root@deep /]# chkconfig --level 2345 sshd on
```

- To start OpenSSH software manually, use the following command:
```
[root@deep /]# /etc/init.d/sshd start
Starting SSHD:            [OK]
```

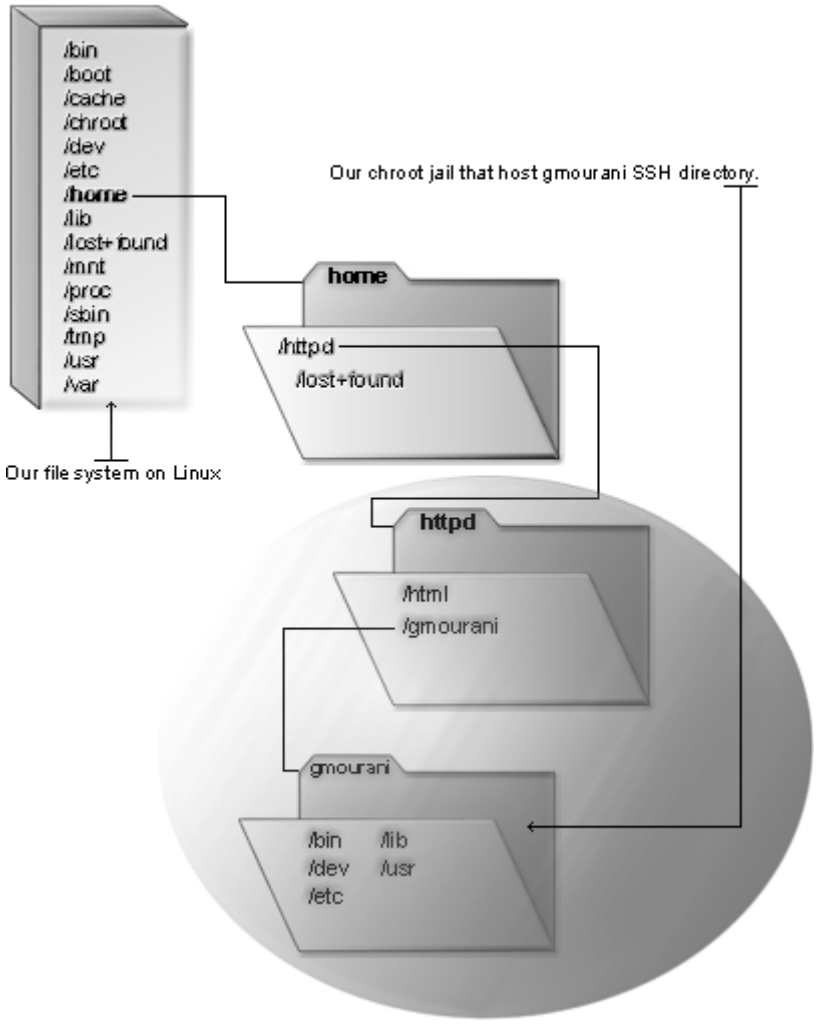
Running OpenSSH in a chroot jail

This section applies only if you want to run OpenSSH in chroot jail environment for some of your users. This kind of setup is useful for web hosting companies that want to provide shell access via remote secure connection with OpenSSH but don't want to allow full access to the server and just limit users to their own web directory. By default, OpenSSH does not support the chroot jail mode and we have to compile it with an external patch to enable the chroot mode extensions. The patch is available from the following site: http://chrootssh.sourceforge.net/

Remember that you have to download the version number equal to the OpenSSH version number you use in order for chroot jail support to work. At the beginning of this chapter, we have already patched the software with the chroot jail mode extensions patch, therefore, we only need to create the required skeleton environment and copy the necessary tools into this chroot jail directory to enable chroot jail support. Below are the steps to follow if you want to run OpenSSH with chroot jail support for the specified users.

The main benefit of a chroot jail is that the jail will limit the portion of the file system the daemon can see to the root directory of the jail. Additionally, since the jail only needs to support OpenSSH, the programs available in the jail can be extremely limited. More importantly, there is no need for setuid-root programs, which can be used to gain root access and break out of the jail.

OpenSSH running in chroot jail



Necessary steps to run OpenSSH in a chroot jail:

What you're essentially doing is creating a skeleton root file system with enough components (binaries, libraries, etc.) to allow Unix to do a chroot when the user logs in.

Step 1

With OpenSSH, it's important to give to your strictly SSH users a real shell account on the Linux system because we want to allow remote shell access, even if limited to running just a few commands on the server.

First, create the new users for this purpose; these users will be the users allowed to connect to your OpenSSH server running in chroot jail mode. These have to be separate from regular user accounts with unlimited access because of how the "chroot" environment works. Chroot makes it appear from the user's perspective as if the level of the file system you've placed them in is the top level of the file system.

Here we create a new SSH user called "gmourani" as an example and set it's the home directory under the /home/httpd/gmourani directory since it is the place where it's the users web directory and web pages will be located.

- Use the following command to create a new SSH user. This step must be done for each additional new user you allow to access your OpenSSH server on OpenNA Linux.

```
[root@deep /]# useradd -m -d /home/httpd/gmourani gmourani

[root@deep /]# passwd gmourani
Changing password for user gmourani
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully
```

- Use the following command to create a new SSH user. This step must be done for each additional new user you allow to access your OpenSSH server on Red Hat Linux.

```
[root@deep /]# useradd -g users -d /home/httpd/gmourani gmourani

[root@deep /]# passwd gmourani
Changing password for user gmourani
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully
```

The useradd command will add the new SSH user called "gmourani" to our Linux server and will set it's the home directory under the /home/httpd/gmourani directory on the system since it is a useful location for remote clients to maintain their web accounts. The passwd command will set the password for this user "gmourani".

Step 2
Once the new SSH users have been created, we must edit the /etc/passwd file and make the appropriated changes to the accounts to allow OpenSSH to chroot when the users login on the system. In general, the sshd daemon will chroot when it encounters the magic token '/./' in a users home directory. Therefore this is what we will add to the passwd file for the SSH user in question.

- Edit the **passwd** file (vi /etc/passwd) and change the following line:

  ```
  gmourani:x:501:100::/home/httpd/gmourani:/bin/bash
  ```

 To read:

  ```
  gmourani:x:501:100::/home/httpd/gmourani/./:/bin/bash
  ```

NOTE: Don't forget to make the same modification for each additional SSH user for whom you want to chroot.

Step 3
Now, we have to create all the necessary chrooted environment subdirectories where we will copy tools we want to allow this SSH user to use on the system.

- Use the following command to create all the necessary chroot subdirectories.

  ```
  [root@deep /]# mkdir /home/httpd/gmourani/bin
  [root@deep /]# mkdir /home/httpd/gmourani/dev
  [root@deep /]# mkdir /home/httpd/gmourani/etc
  [root@deep /]# mkdir /home/httpd/gmourani/lib
  [root@deep /]# mkdir /home/httpd/gmourani/usr
  [root@deep /]# mkdir /home/httpd/gmourani/usr/bin
  [root@deep /]# mkdir /home/httpd/gmourani/usr/lib
  ```

- For Red Hat Linux 7.3 users, you should create the following additional directory:

  ```
  [root@deep /]# mkdir /home/httpd/gmourani/lib/i686
  ```

Step 4
Next, we must change the permissions on all the chroot glue subdirectories to mode (0111/d--x--x--x) for security reasons.

- Use the following command to change the permissions of all the subdirectories.

  ```
  [root@deep /]# chmod -R 0111 /home/httpd/gmourani/*
  ```

Step 5
Once all permissions of the supporting glues have been changed, it is time to copy the required binary programs to the related subdirectories in the chroot area for OpenSSH to work.
These programs are necessary to allow the SSH users to list, create directory, copy, remove, and edit files on the SSH chroot jail directory. If there are features you don't want the user to be able to use, then don't copy them to the chroot area.

- Use the following commands to copy the require binaries programs into the chroot area.

```
[root@deep /]# cp /bin/bash /home/httpd/gmourani/bin/
[root@deep /]# cp /bin/cp /home/httpd/gmourani/bin/
[root@deep /]# cp /bin/ls /home/httpd/gmourani/bin/
[root@deep /]# cp /bin/mkdir /home/httpd/gmourani/bin/
[root@deep /]# cp /bin/grep /home/httpd/gmourani/bin/
[root@deep /]# cp /bin/rm /home/httpd/gmourani/bin/
[root@deep /]# cp /bin/vi /home/httpd/gmourani/bin/
[root@deep /]# cp /usr/bin/dircolors /home/httpd/gmourani/usr/bin/
[root@deep /]# chmod 0111 /home/httpd/gmourani/bin/*
[root@deep /]# chmod 0111 /home/httpd/gmourani/usr/bin/*
```

NOTE: The above chmod commands will change default permissions of those programs under the /bin directories of the chroot jail area to be (0111 ---x--x–x) because we don't want users to be able to modify or read binaries in the chroot area but just to execute them if necessary.

Step 6
The binaries we have copied into the chroot area have been compiled with shared libraries by default and for this reason it is important to find the shared libraries dependencies associated with them and copy them into the /lib subdirectory in the chroot jail area that we created earlier.

To find the shared library dependencies of binaries, you have to use the ldd command of Linux. You must copy all the libraries below to the /home/httpd/gmourani/lib directory of the chroot area. These libraries are part of libc, and needed by various programs.

- Use the following commands to copy the require libraries into the chroot area.

```
[root@deep /]# cp /lib/libtermcap.so.2 /home/httpd/gmourani/lib/
[root@deep /]# cp /lib/libdl.so.2 /home/httpd/gmourani/lib/
[root@deep /]# cp /lib/libc.so.6 /home/httpd/gmourani/lib/
[root@deep /]# cp /lib/libgcc_s.so.1 /home/httpd/gmourani/lib/
[root@deep /]# cp /lib/ld-linux.so.2 /home/httpd/gmourani/lib/
[root@deep /]# cp /usr/lib/libncurses.so.5 /home/httpd/gmourani/usr/lib/
[root@deep /]# strip -R .comment /home/httpd/gmourani/lib/*
```

- For Red Hat Linux 7.3 users, you should copy the following additional library:

```
[root@deep /]# cp /lib/i686/libc.so.6 /home/httpd/gmourani/lib/i686/
```

WARNING: Depending on what has been compiled, the required shared libraries may be different than the ones illustrated above. Please use the `ldd` command on each binary under `/bin` subdirectories of the chroot jail to find out the ones you need and copy them to the `/lib` subdirectory of the chroot area.

The "`strip -R .comment`" command will remove all the named section "`.comment`" from the libraries files under the `/lib` subdirectory and will make them smaller in size and can help in the performance of them.

Step 7
One of the last step to do, is to make a copy of the "`DIR_COLORS`" and "`passwd`" files located under the `/etc` directory to our chroot jail for `SSH` to be able to find it.

- Use the following commands to copy the file into the chroot area.
```
[root@deep /]# cp /etc/DIR_COLORS /home/httpd/gmourani/etc/
[root@deep /]# cp /etc/passwd /home/httpd/gmourani/etc/
```

Step 8
Finally, we have to create the `/home/httpd/gmourani/dev/null` device file and set its mode appropriately.

- Use the following commands to create the `null` device into the chroot area.
```
[root@deep /]# mknod /home/httpd/gmourani/dev/null c 1 3
[root@deep /]# chmod 666 /home/httpd/gmourani/dev/null
```

Creating `OpenSSH` private & public keys

This section deals with actions that need to be performed to create new private/public keys for our users to establish a secure connection on the server. There are cryptosystems where encryption and decryption are done using separate keys, and it is not possible to derive the decryption key from the encryption key. The idea is that each user creates a public/private key pair for authentication purposes. The server knows the public key, and only the user knows the private key.

The file `$HOME/.ssh/authorized_keys` lists the public keys that are permitted for logging in. When the user logs in, the `ssh` program tells the server which key pair it would like to use for authentication. The server checks if this key is permitted, and if so, sends the user (actually the `ssh` program running on behalf of the user) a challenge, a random number, encrypted by the user's public key. The challenge can only be decrypted using the proper private key. The user's client then decrypts the challenge using the private key, proving that he/she knows the private key but without disclosing it to the server.

Step 1
Below, are the steps to follow to create a new `SSH` private & public key for one user. This example assumes that secure encrypted connections will be made between Linux servers.

- To create your (`RSA`) private & public keys for `SSH2` of LOCAL, use the commands:
```
[root@deep /]# su gmourani
[gmourani@deep /]$ ssh-keygen -t rsa
Generating public/private rsa key pair.
```

```
Enter file in which to save the key (/home/gmourani/.ssh/id_rsa):
Created directory '/home/gmourani/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/gmourani/.ssh/id_rsa.
Your public key has been saved in /home/gmourani/.ssh/id_rsa.pub.
The key fingerprint is:
ba:0c:08:6d:9d:51:4f:b3:32:68:9b:0d:83:ce:be:bd gmourani@deep
```

WARNING: The above example assumes that you want to generate (RSA) private & public keys for SSH protocol 2 (highly recommended). If you want to generate (RSA) private & public keys for SSH protocol 1, then you must use the '-t rsa1' option to the key generation command as follows:

```
[root@deep /]# su gmourani
[gmourani@deep /]$ ssh-keygen -t rsa1
```

Using the '-t rsa1' option will generate SSH1 instead of SSH2 private & public keys. The SSH1 private key will be named "identity" and the public key will be "identity.pub". The '-t' option is used to specify the type of the key to create. The possible values are "rsa1" for protocol version 1 and "rsa" or "dsa" for protocol version 2.

If you have multiple accounts you might want to create a separate key on each of them. You may want to have separate keys for:

- Your server (1)
- Your server (2)
- Your server (3)

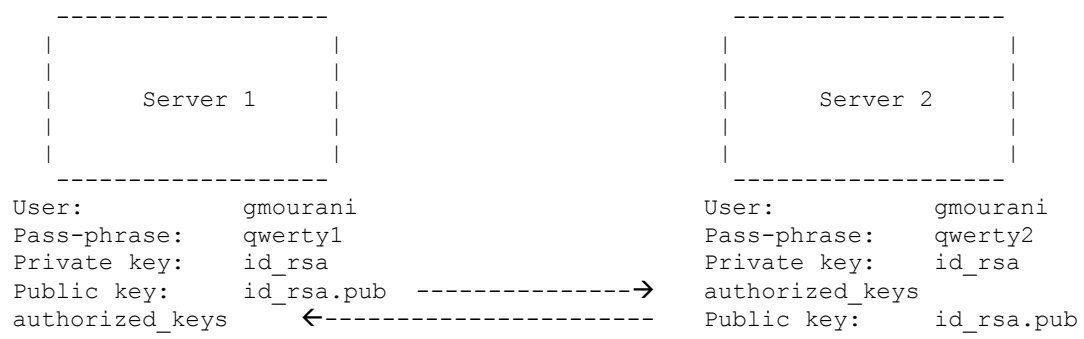
This allows you to limit access between these servers, e.g. not allowing the first server (1) account to access your second server (2) account or the third server (3). This enhances the overall security in the case any of your authentication keys are compromised for any reason.

Step 2
Copy your local public key **id_rsa.pub** for SSH2 or **identity.pub** for SSH1 from the /home/gmourani/.ssh directory remotely under the name, say, "authorized_keys". One way to copy the file is to use the ftp command or you might need to send your public key in electronic mail to the administrator of the other system. Just include the contents of the ~/.ssh/id_rsa.pub or ~/.ssh/identity.pub file in the message.

To resume the required steps:

1) The user creates his/her RSA keys pair by running ssh-keygen. This stores the private key in id_rsa (SSH2) or in identity (SSH1) and the public key in id_rsa.pub (SSH2) or in identity.pub (SSH1) into the user's home directory on the LOCAL machine.

2) The user should then copy the id_rsa.pub key (SSH2) or identity.pub key (SSH1) to $HOME/.ssh/authorized_keys into his/her home directory on the REMOTE machine.



Public key of user `gmourani` on the first server (1) is sending to the second server (2) under the $HOME directory of user `gmourani` and become '`authorized_keys`'; the same action is made on the second server (2). The public key of user `gmourani` on server (2) is sending to server (1) under the $HOME directory of user `gmourani` and become '`authorized_keys`'.

NOTE: `OpenSSH`'s public key is a one-line string. Adding public keys from commercial `SSH` tools which stretch the public key over several lines, will not be recognized by `OpenSSH`.

`OpenSSH` Users Tools

The commands listed below are some that we use regularly, but many more exist, and you should check the manual pages and documentation of `OpenSSH` for more details.

`ssh`

The `ssh` (**S**ecure **Sh**ell) command provides secure encrypted communications between two untrusted hosts over an insecure network. It is a program for securely logging into a remote machine and executing commands from there. It is a suitable replacement for insecure programs like `telnet`, `rlogin`, `rcp`, `rdist`, and `rsh`.

- To login to a remote machine, use the following command:
 `[root@deep /]#` **`ssh -l <login_name> <hostname>`**

 For example:
  ```
  [root@deep /]# ssh -l gmourani deep.openna.com
  gmourani@deep.openna.com's password:
  Last login: Tue Oct 19 1999 18:13:00 -0400 from deep.openna.com
  No mail.
  [gmourani@deep gmourani]$
  ```

Where <`login_name`> is the name you use to connect to the `ssh` server and <`hostname`> is the remote address (you can use `IP` address here) of your `ssh` server.

scp

The scp (**S**ecure **C**o**p**y) utility copies files from the local system to a remote system or vice versa, or even between two remote systems using the scp command.

- To copy files from remote to local system, use the following commands:

```
[root@deep /]# su gmourani
[gmourani@deep /]$ scp -p <login_name@hostname>:/dir/for/file
localdir/to/filelocation
```

For example:

```
[gmourani@deep /]$ scp -p gmourani@mail:/etc/test1 /tmp
Enter passphrase for RSA key 'gmourani@mail.openna.com':
test1            |          2 KB |   2.0 kB/s | ETA: 00:00:00 | 100%
```

- To copy files from local to remote system, use the following commands:

```
[root@deep /]# su gmourani
[gmourani@deep /]$ scp -p localdir/to/filelocation
<username@hostname>:/dir/for/file
```

For example:

```
[gmourani@deep /]$ scp -p /usr/bin/test2 gmourani@mail:/var/tmp
gmourani@mail's password:
test2            |          7 KB |   7.9 kB/s | ETA: 00:00:00 | 100%
```

WARNING: The "-p" option indicates that the modification and access times, as well as modes of the source file, should be preserved on the copy. Usually this is desirable. Please check the chapter related to backups in this book for more information about other possible uses of SSH technology with Linux.

Changing your pass-phrase

You can change the pass-phrase at any time by using the -p option of ssh-keygen.

- To change the pass-phrase, use the following commands:

```
[root@deep /]# su gmourani
[gmourani@deep /]$ ssh-keygen -p
Enter file in which the key is (/home/gmourani/.ssh/id_rsa):
Enter old passphrase:
Key has comment '/home/gmourani/.ssh/id_rsa'
Enter new passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved with the new passphrase.
```

Further documentation

For more details, there are several manual pages about OpenSSH that you can read:

```
$ man ssh (1)           - OpenSSH secure shell client (remote login program).
$ man ssh [slogin] (1)  - OpenSSH secure shell client (remote login program).
$ man ssh-add (1)       - Adds identities for the authentication agent.
$ man ssh-agent (1)     - Authentication agent.
$ man ssh-keygen (1)    - Authentication key generation.
$ man sshd (8)          - Secure shell daemon.
$ sftp-server (8)       - SFTP server subsystem.
```

CHAPTER 17

Sudo

IN THIS CHAPTER

1. Compiling - Optimizing & Installing `Sudo`
2. Configuring `Sudo`
3. A more complex `sudoers` configuration file
4. Securing `Sudo`
5. `Sudo` Users Tools

Linux Sudo

Abstract

Sudo (**su**peruser **do**) is a security program designed to allow a system administrator to give limited root privileges to users and log root activity. The basic philosophy is to give as few privileges as possible, but still allow people to get their work done. It operates on a per-command basis and it is not a replacement for the shell. This means that you have to use it every time you need to execute some commands with "root" privilege on the server.

In general, it does the same function as the command 'su' does on the Linux but with a big difference that we have full control about what should be done by which users, what commands a user may run, etc. Here is some of its feature:

- ✓ It provides ability to restrict what commands a user may run on a per-host basis.
- ✓ It does copious logging of each command, providing a clear audit trail.
- ✓ It can log all commands to a central host (as well as on the local host).
- ✓ It uses timestamp files to implement a "ticketing" system "root" time access.
- ✓ Its configuration file is setup in such a way that you could use it on many machines.

Imagine that your boss asks you to create a new account for the new webmaster of your company. This webmaster will be responsible of the web server, but you don't know if this person will stay with the company for a long time or not. You don't want to give him full "root" privileges via the 'su' command of Linux because you don't trust him or he doesn't need to have full "root" access to manage a web server. This is where a program like sudo will help you to same time and protect your server.

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest Sudo version number is 1.6.6

The procedures given in this chapter are likely to work on all Linux platforms, but we have only tested it on OpenNA Linux and Red Hat Linux.

Packages

Please check http://www.sudo.ws/ regularly for the latest status. We chose to install from source because it provides the facility to fine tune the installation.

Source code is available from:
Sudo Homepage: http://www.sudo.ws/
Sudo FTP Site: 128.138.243.20
You must be sure to download: sudo-1.6.6.tar.gz

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all the files installed onto the system if you want to update the package in the future. To solve this problem, it's a good idea to make a list of files on the system before you install Sudo, and then one afterwards, and then compare them using the **diff** utility to find out what files were placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep root]# find /* > Sudo1
  ```
- And the following one after you install the software:
  ```
  [root@deep root]# find /* > Sudo2
  ```
- Then use this command to get a list of what changed:
  ```
  [root@deep root]# diff Sudo1 Sudo2 > Sudo-Installed
  ```

Compiling - Optimizing & Installing Sudo

Below are the steps that you must make to configure, compile and optimize the Sudo software before installing it on your system. First off, we install the program as user 'root' so as to avoid any authorization problems.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- This can be done with the following commands:
  ```
  [root@deep /]# cp sudo-version.tar.gz /var/tmp/
  [root@deep /]# cd /var/tmp/
  [root@deep tmp]# tar xzpf sudo-version.tar.gz
  ```

Step 2
Next, move into the newly created Sudo source directory and perform the following steps to configure and optimize the software for your system.

- To move into the newly created Sudo source directory use the command:
  ```
  [root@deep tmp]# cd sudo-1.6.6/
  ```
- To configure and optimize Sudo use the following compilation lines:
  ```
  CFLAGS="-O2 -march=i686 -funroll-loops"; export CFLAGS
  ./configure \
  --prefix=/usr \
  --sbindir=/usr/sbin \
  --with-logging=syslog \
  --with-logfac=authpriv \
  --with-pam \
  --with-env-editor \
  --with-ignore-dot \
  --with-tty-tickets \
  --disable-root-mailer \
  --disable-root-sudo \
  --disable-path-info \
  --with-mail-if-noperms \
  --with-mailsubject="*** Sudo SECURITY information for %h ***"
  ```

This tells Sudo to set itself up for this particular configuration setup with:

- Log sudo messages via syslog.
- The syslog facility to log with is authpriv.
- Enable PAM support.
- Use the environment variable EDITOR for visudo.
- Ignore '.' in the PATH.
- Use a different ticket file for each user/tty combo.
- Don't run the mailer as root, run as the user since it's safer.
- Don't allow root to run sudo command.
- Print 'command not allowed' instead of 'command not found'.
- Send mail to sysadmin if user not allowed to runs command.
- Change subject of sudo mail result.

Step 3
Now, we must make a list of all files on the system before installing the software, and one afterwards, then compare them using the **diff** utility to find out what files are placed where and finally we install the Sudo software:

```
[root@deep sudo-1.6.6]# make
[root@deep sudo-1.6.6]# cd
[root@deep root]# find /* > Sudo1
[root@deep root]# cd /var/tmp/sudo-1.6.6/
[root@deep sudo-1.6.6]# make install
[root@deep sudo-1.6.6]# strip /usr/bin/sudo
[root@deep sudo-1.6.6]# strip /usr/sbin/visudo
[root@deep sudo-1.6.6]# mkdir –p –m0700 /var/run/sudo
[root@deep sudo-1.6.6]# cd
[root@deep root]# find /* > Sudo2
[root@deep root]# diff Sudo1 Sudo2 > Sudo-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

Step 4
Once the configuration, optimization, compilation, and installation of the Sudo software have been accomplished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete Sudo and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf sudo-version/
[root@deep tmp]# rm -f sudo-version.tar.gz
```

Configuring Sudo

After Sudo has been built and installed successfully in your system, your next step is to configure and customize its configuration files.

- ✓ /etc/sudoers: (The Sudo Configuration File)
- ✓ /etc/pam.d/sudo: (The Sudo PAM Support Configuration File)

/etc/sudoers: The Sudo Configuration File

The /etc/sudoers file is the main configuration file for Sudo. It is in this configuration file that Sudo gets all of its information on the way it should run on your system. The parameters entered in the sudoers configuration file will decide how regular users should use sudo to get "root" privileges to accomplish their works.

On production servers where shell access and "root" privilege are limited to just some trusted regular users, the sudoers configuration file should be very simple to configure. All we need is to define a group under which trusted people are allowed to run all commands as "root" and put the related people into this group name. This kind of configuration file works a little bit like PAM to control who can have "root" access to the system via the 'su' command but in a more secure and natural way.

In the sudoers configuration file below, we will show you the correct setup to limit some users of a specific group name (wheel) to sudo and get "root" access on the system. This is the most used configuration file for the majority of users. We only want to allow some regular users with shell access on the server to sudo to "root" account.

Later, we will explain how to configure the sudoers file on server where many shell accesses are available for users. These kinds of servers are generally development servers where developers work and need special "root" privileges depending on their work and tasks to accomplish.

Finally, I will inform you that with sudo, we must edit the sudoers configuration file with the "visudo" program which comes installed on your system for this purpose. Never edit the sudoers file with other editor like "vi", always use the "visudo" command when you want to change information inside the sudoers configuration file.

Step1
Ok, it's time to configure sudoers to allow users who are members of the group "wheel" to get "root" access on the server. First, we have to edit sudoers and make the changes.

- Edit the **sudoers** file (visudo) and set your needs. Below is what we recommend you use for production servers:

```
# This file MUST be edited with the 'visudo' command as root.

# Defaults specification
Defaults        rootpw

# User privilege specification
# Super-user root can run anything as any user.
root            ALL=(ALL) ALL

# Comment if you don't want to allow people in group wheel to
# run all commands as super-user root.
%wheel          ALL=(ALL) ALL
```

This tells the `sudoers` file to set itself up for this particular configuration with:

`Defaults rootpw`
With `sudo`, certain configuration options may be changed from their default values at runtime via one or more "`Default_Entry`" options defined in the `sudoers` file. This is what we do here. In our case, we inform `sudo` with the "`Defaults rootpw`" option that we want it to prompt any allowed user who wants to `sudo` to super-user "`root`" to enter the "`root`" password instead of the password of the invoking user.

By default `sudo` asks for the users password instead of super-user password when someone wants to `sudo` to "`root`" user. Because in this configuration file we want to allow full "`root`" access for users that are members of the "`wheel`" group and because we trust these users, we decide to change the default `sudo` setting and ask for "`root`" password before having access to "`root`" privilege on the server.

This setting is useful when you make secure remote connections on the server with `SSH` software and want to `sudo` to "`root`" user.

`root ALL=(ALL) ALL`
Here we inform `sudo` that the super-user "`root`" can run anything as any user on the server. This option is required for the software to work or there is no need to use `sudo`. The "`ALL=(ALL) ALL`" parameter means that everything is allowed for the super-user "`root`".

`%wheel ALL=(ALL) ALL`
Here we allow every user who is a member of the group "`wheel`" to run all commands as super-user "`root`" on the system. This is the equivalence of what we achieve with the `PAM` security feature, but in a most efficient and secure way.

Step 2
Once the `sudoers` file has been configured, it is time to add some users who will be allowed to `sudo` to "`root`" account.

- If you want to make, for example, the user "`sysadmin`" a member of the "`wheel`" group, and thus be able to `sudo` to "`root`", use the following command:

```
[root@deep /]# usermod -G10 sysadmin
```

This means "`G`" is a list of supplementary groups that the user is also a member of. "`10`" is the numeric value of the user's ID "`wheel`", and "`sysadmin`" is the user we want to add to the "`wheel`" group. Use the same command above for all users on your system you want to be able to `sudo` to "`root`" account.

/etc/pam.d/sudo: The Sudo PAM Support Configuration File

For better security of Sudo, we have compiled it to use the PAM mechanism for password authentication.

Step 1
To be able to use this feature, we must create the /etc/pam.d/sudo file and add the following parameters inside it.

- Create the **sudo** file (touch /etc/pam.d/sudo) and add the following lines:

```
#%PAM-1.0
auth       required     /lib/security/pam_stack.so service=system-auth
account    required     /lib/security/pam_stack.so service=system-auth
password   required     /lib/security/pam_stack.so service=system-auth
session    required     /lib/security/pam_stack.so service=system-auth
```

Step2
Now, set the permissions of the sudo file to be (0640/-rw-r-----) and owned by the super-user 'root' for security reasons.

- To change the permissions and ownership of the sudo file, use the commands:

```
[root@deep /]# chmod 640 /etc/pam.d/sudo
[root@deep /]# chown 0.0 /etc/pam.d/sudo
```

A more complex sudoers configuration file

For those who want to have complete control on who can use the "root" account on the server when there are many users with shell access doing different tasks, here is the sudoers configuration file to go with. As you'll see it is more complex and covers some basic definitions.

It is important to understand how the sudo policy works. To be as clear as possible, I will simply say that when you allow full access to user with the "ALL" definition, you cannot deny other access or privileges. Therefore, the best way is to allow only what you want user to be able to run with "root" privilege through the "Cmnd alias specification" part of the configuration file and use the defined aliases rules under the "User privilege specification" part of the configuration file. Here is a working sudoers configuration file to better understand what I mean.

- Edit the **sudoers** file (visudo) and set your needs. Below is what we recommend you use for servers that have many users with shell access:

```
# /etc/sudoers: OpenNA, Inc. (last updated 2002 Apr 19)
# This file MUST be edited with the 'visudo' command as root.

# User alias specification
User_Alias      FULLTIME_USERS = sysadmin, gmourani
User_Alias      PARTTIME_USERS = zeljko, mary

# Cmnd alias specification
Cmnd_Alias      HTTP = /etc/init.d/httpd, /bin/vi /etc/httpd/*
Cmnd_Alias      FTP  = /etc/init.d/proftpd, /bin/vi /etc/proftpd.conf
Cmnd_Alias      SMTP = /etc/init.d/exim, /bin/vi /etc/mail/*
Cmnd_Alias      SQL  = /etc/init.d/mysqld, /usr/bin/mysql,
/usr/bin/mysqladmin
Cmnd_Alias      BIND = /etc/init.d/named, /bin/vi /chroot/named/*
```

```
# Defaults specification
Defaults:FULLTIME_USERS         rootpw
Defaults:FULLTIME_USERS         !lecture

# User privilege specification
# Super-user root can run anything as any user.
root            ALL=(ALL) ALL

# Every users member of the group wheel will be allowed
# to run all commands as super-user root.
%wheel          ALL=(ALL) ALL

# Full time users may run anything but need a password.
FULLTIME_USERS  ALL = ALL

# Part time users may administrate httpd, ftpd, smtp, sql, and bind
servers.
PARTTIME_USERS  ALL = HTTP, FTP, SMTP, SQL, BIND
```

This tells the `sudoers` file to set itself up for this particular configuration with:

```
User_Alias      FULLTIME_USERS = sysadmin, gmourani
User_Alias      PARTTIME_USERS = zeljko, mary
```

The "`User_Alias`" definition lines are used to define local users who may `sudo` to the "`root`" account on the server. The definition works on the following way:

```
Alias_Type NAME = item1, item2, ...
```

Where "`Alias_Type`" is the type of alias to use, in our case, we use "`User_Alias`" to define local users aliases on the system. A `NAME` is a string of uppercase letters, numbers, or underscores characters ('_'). A `NAME` must always start with an uppercase letter. You can use as any name as you like to define the `NAME`.

In our example, we use "`FULLTIME_USERS`" to define local users on the system who have full time access to the server and "`PARTTIME_USERS`" to define local users on the server who have part time access to the server for different reason. Item represents usernames to add in each category separated by [`,`].

NOTE: It is important to note that users added to the "`User_Alias`" definition will be able to `sudo` to super-user "`root`" account even if their names do not appear under the group "`wheel`".

```
Cmnd_Alias HTTP = /etc/init.d/httpd, /bin/vi /etc/httpd/*
Cmnd_Alias FTP  = /etc/init.d/proftpd, /bin/vi /etc/proftpd.conf
Cmnd_Alias SMTP = /etc/init.d/exim, /bin/vi /etc/mail/*
Cmnd_Alias SQL  = /etc/init.d/mysqld, /usr/bin/mysql, /usr/bin/mysqladmin
Cmnd_Alias BIND = /etc/init.d/named, /bin/vi /chroot/named/*
```

Here we use another "`Alias_Type`" definition. This type of alias is used to define commands, programs, files, and directories that we want to allow certain users to be able to use when issuing the `sudo` command on the system. The definition works on the following way:

```
Alias_Type NAME = item1, item2, ...
```

Where "Alias_Type" is the type of alias to use, in our case, we use "Cmnd_Alias" to define command aliases on the system. A NAME is a string of uppercase letters, numbers, or underscores characters ('_'). A NAME must always start with an uppercase letter. You can use any name you like to define the NAME.

In our example, we use "HTTP, FTP, SMTP, SQL, and BIND" to define command names aliases associated with the commands we will allow local users to run when issuing the sudo command on the system. Item represent the commands, files, programs, or directories to add in each category separated by [,].

```
Defaults:FULLTIME_USERS         rootpw
Defaults:FULLTIME_USERS         !lecture
```

With sudo, certain configuration options may be changed from their default values at runtime via one or more "Default_Entry" options. Again, this is what we do here. In our case, we inform sudo with the "Defaults:FULLTIME_USERS rootpw" option that we want it to prompt any users with full time access on the server "FULLTIME_USERS" to enter the "root" password instead of their own password.

Remember that by default sudo asks for the users password instead of super-user password when someone wants to sudo to "root". Because we allow full "root" access for users under the "FULLTIME_USERS" alias, we decide to change the default sudo setting and ask for the "root" password before giving access to "root" privileges on the server. This also means that users under the "PARTTIME_USERS" alias will have to enter their own password and not the "root" password.

This is a security feature to separate trusted users with full time access on the server from semi-trusted users with part time access on the system. Users having part time access could be students, limited administrators, or anything else that you think about. In this way, users with part time access do not know the "root" password since they enter their own passwords to get some "root" privileges and we don't need to change "root" password every time these users leave the company.

The second "Default_Entry" option "Defaults:FULLTIME_USERS !lecture", simply informs sudo to not send a short lecture message about the policy use of the program to users with full time access on the server.

```
FULLTIME_USERS  ALL = ALL
```

Here we allow every user who is listed in the "FULLTIME_USERS" alias of the configuration file to run all commands as super-user "root" on the system. This is the equivalence of what we achieve with the PAM security feature.

```
PARTTIME_USERS  ALL = HTTP, FTP, SMTP, SQL, BIND
```

Here we allow every user who is listed in the "PARTTIME_USERS" alias of the configuration file to only run the specified commands as super-user "root". These users will be able to edit, run and restart the specified services and nothing else. Therefore, they only have limited "root" access on the server to accomplish their tasks.

Securing Sudo

This section deals specifically with actions we can take to improve and tighten security under Sudo. Sudo is very good and well-written software with high security in mind. Once properly compiled, installed, and configured, there are only some little things that we can do to better secure it. Most important of all the security measures are already made within the software.

Disabling su access on the server

Once sudo is configured and running on your server, you really don't need to continue to use 'su' to get "root" access on the system. Here, I show you how to disable 'su' command to provide "root" access. The simpler approach will be to remove the SUID bit on the 'su' program. In this way, no one will be able to use it to gain "root" anymore.

- To remove the SUID bit on the 'su' command, use the following command:

```
[root@deep /]# chmod 0511 /bin/su
```

Sudo Users Tools

The commands and options listed bellows are some that we use regularly, but many more exist, and you should check the manual pages and documentation of Sudo for more details.

Running sudo for user with full "root" access

Basically, sudo is used by prepending "sudo" (followed by a space) to your command. It then prompts you for your personal password or root password depending of the configuration and then checks the /etc/sudoers configuration file to make sure you have "sudo" permission to run that command on the server. Sudo runs the command as root (or another user) and logs the details to syslog. It also logs problems and other invalid uses.

When you have full "root" privileges on the system because you are listed in the sudoers file as one user with all "root" access right, you can run the sudo command with the -s (shell) option to become the super-user "root" on the server.

- To sudo as super-user "root" with shell access, use the following command:

```
[sysadmin@deep /]# sudo -s
Password:
```

To be able to use the above command, you should have all "root" access rights in the sudoers configuration file. Please note that, in this example, you have to enter the super-user "root" password and not the password of the user "sysadmin".

Running `sudo` for user with limited "`root`" access

If you are a user with limited "`root`" access rights on the server, you cannot use the previous `sudo` command to `sudo` as super-user "`root`" on the server. Instead, you should prepend the `sudo` command with the command you want to run as "`root`" user. This supposes that commands you expect to run are listed as allowed commands to use inside the `sudoers` configuration file.

- To `sudo` as super-user "`root`" with limited access, use the following command:
```
[mary@deep /]# sudo /etc/init.d/httpd restart
Password:
```

The above command will restart the `httpd` web server daemon on the system even if you are the user called "`mary`" because the `sudoers` file allows you to do it as super-user "`root`". Please note that in this example you have to enter the password of user "`mary`" and not the password of the super-user "`root`".

Further documentation

For more details, there are some manual pages you can read:

```
$ man sudoers (5)       - List of which users may execute what.
$ man sudo (8)          - Execute a command as another user.
$ man visudo (8)        - Used to edit the sudoers file.
```



Part V

Monitoring & System Integrity

CHAPTER 18

sXid

IN THIS CHAPTER

1. Compiling - Optimizing & Installing `sXid`
2. Configuring `sXid`
3. `sXid` Administrative Tools

Linux sXid

Abstract

SUID/SGID files can be a security hazard. To reduce the risks, we have previously removed the 's' bits from root-owned programs that won't require such privileges (See chapter related to General System Security), but future and existing files may be set with these 's' bits enabled without you being notified.

sXid is an all in one suid/sgid monitoring program designed to be run by "cron" on a regular basis. Basically, it tracks any changes in your s[ug]id files and folders. If there are any new ones, ones that aren't set any more, or they have changed bits or other modes then it reports the changes in an easy to read format via email or on the command line. sXid will automate the task to find all SUID/SGID on your server and report them to you. Once installed you can forget it and it will do the job for you.

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest sXid version number is 4.0.2

The procedures given in this chapter are likely to work on all Linux platforms, but we have only tested it on OpenNA Linux and Red Hat Linux.

Packages

The following is based on information listed by sXid as of 2002/06/24. Please check ftp://marcus.seva.net/pub/sxid/ regularly for the latest status. We chose to install from source because it provides the facility to fine tune the installation.

Source code is available from:
sXid Homepage: ftp://marcus.seva.net/pub/sxid/
sXid FTP Site: 137.155.111.51
You must be sure to download: sxid_4.0.2.tar.gz

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all the files installed onto the system if you want to update the package in the future. To solve this problem, it's a good idea to make a list of files on the system before you install sXid, and then one afterwards, and then compare them using the **diff** utility to find out what files were placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep root]#  find /* > sXid1
  ```
- And the following one after you install the software:
  ```
  [root@deep root]#  find /* > sXid2
  ```
- Then use the following command to get a list of what changed:
  ```
  [root@deep root]#  diff sXid1 sXid2 > sXid-Installed
  ```

Compiling - Optimizing & Installing sXid

Below are the steps that you must make to configure, compile and optimize the sXid software before installing it on your system. First off, we install the program as user 'root' so as to avoid any authorization problems.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- This can be done with the following commands:
  ```
  [root@deep /]# cp sxid_version.tar.gz /var/tmp/
  [root@deep /]# cd /var/tmp/
  [root@deep tmp]# tar xzpf sxid_version.tar.gz
  ```

Step 2
Now move into the newly created sXid source directory and perform the following steps to configure and optimize the software for your system.

- To move into the newly created sXid directory use the following command:
  ```
  [root@deep tmp]# cd sxid-4.0.2/
  ```
- To configure and optimize sXid use the following compilation lines:
  ```
  CFLAGS="-O2 -march=i686 -funroll-loops"; export CFLAGS
  ./configure \
  --prefix=/usr \
  --sysconfdir=/etc \
  --mandir=/usr/share/man
  ```

WARNING: Pay special attention to the compile CFLAGS line above. We optimize sXid for an i686 CPU architecture with the parameter "-march=i686". Please don't forget to adjust this CFLAGS line to reflect your own system.

Step 3
Now, we must make a list of all files on the system before installing the software, and one afterwards, then compare them using the **diff** utility to find out what files are placed where and finally we install the sXid software:

```
[root@deep sXid-4.0.2]# cd
[root@deep root]# find /* > sXid1
[root@deep root]# cd /var/tmp/sxid-4.0.2/
[root@deep sxid-4.0.2]# make install
[root@deep sxid-4.0.2]# cd
[root@deep root]# find /* > sXid2
[root@deep root]# diff sXid1 sXid2 > sXid-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

Step 4
Once the configuration, optimization, compilation, and installation of the sXid software have been accomplished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete sXid and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf sxid-version/
[root@deep tmp]# rm -f sxid_version_tar.gz
```

The rm command as used above will remove all the source files we have used to compile and install sXid. It will also remove the sXid compressed archive from the /var/tmp directory.

Configuring sXid

After sXid has been built and installed successfully in your system, your next step is to configure and customize its configuration files to fit your needs.

- ✓ /etc/sxid.conf: (The sXid Configuration File)
- ✓ /etc/cron.daily/sxid: (The sXid Cron File)

/etc/sxid.conf: The sXid Configuration File

The configuration file for sXid allows you to set options that modify the operation of the program. It is well commented and very basic.

Step 1
We must change the default one to fit our requirements and operating system. The text in bold are the parts of the configuration file that must be customized and adjusted to satisfy our needs.

- Edit the **sxid.conf** file (vi /etc/sxid.conf) and set your needs. Below is what we recommend you.

```
# Configuration file for sXid
# Note that all directories must be absolute with no trailing /'s

# Where to begin our file search
SEARCH = "/"
```

```
# Which subdirectories to exclude from searching
EXCLUDE = "/proc /mnt /cdrom /floppy"

# Who to send reports to
EMAIL = "root"

# Always send reports, even when there are no changes?
ALWAYS_NOTIFY = "no"

# Where to keep interim logs. This will rotate 'x' number of
# times based on KEEP_LOGS below
LOG_FILE = "/var/log/sxid.log"

# How many logs to keep
KEEP_LOGS = "5"

# Rotate the logs even when there are no changes?
ALWAYS_ROTATE = "no"

# Directories where +s is forbidden (these are searched
# even if not explicitly in SEARCH), EXCLUDE rules apply
FORBIDDEN = "/home /tmp"

# Remove (-s) files found in forbidden directories?
ENFORCE = "yes"

# This implies ALWAYS_NOTIFY. It will send a full list of
# entries along with the changes
LISTALL = "no"

# Ignore entries for directories in these paths
# (this means that only files will be recorded, you
# can effectively ignore all directory entries by
# setting this to "/"). The default is /home since
# some systems have /home g+s.
IGNORE_DIRS = "/home"

# File that contains a list of (each on it's own line)
# other files that sxid should monitor. This is useful
# for files that aren't +s, but relate to system
# integrity (tcpd, inetd, apache...).
# EXTRA_LIST = "/etc/sxid.list"

# Mail program. This changes the default compiled in
# mailer for reports. You only need this if you have changed
# it's location and don't want to recompile sxid.
MAIL_PROG = "/bin/mail"
```

Step 2
Now, for security reasons, change the mode of this file to be `0400`.

- This can be done with the following command:
 `[root@deep /]#` **`chmod 400 /etc/sxid.conf`**

/etc/cron.daily/sxid: The sXid Cron File

The sxid file is a small script executed automatically by the "crond" program of your server each day to tracks any changes in your s[ug]id files and folders.

Step 1
If there are any new ones, ones that aren't set any more, or they have changed bits or other modes then it reports the changes. If you intend to automate this task, follow the simple steps below.

- Create the **sxid** script file (touch /etc/cron.daily/sxid) and add the following lines:

```
#!/bin/sh

SXID_OPTS=

if [ -x /usr/bin/sxid ]; then
        /usr/bin/sxid ${SXID_OPTS}
fi
```

Step2
Now, make this script executable and change its permissions to be 0510.

- This can be done with the following command:
 `[root@deep /]# chmod 510 /etc/cron.daily/sxid`

sXid Administrative Tools

After your desired configuration options have been set and the program is running, we can play with its utility. The sXid software is meant to run as a cronjob. It must run once a day, but busy shell boxes may want to run it twice a day. You can also run this manually for spot-checking.

- To run sXid manually, use the command:

```
[root@deep /]# sxid -k
sXid Vers  : 4.0.2
Check run  : Thu Apr 25 19:35:36 2002
This host  : deep.openna.com
Spotcheck  : /root
Excluding  : /proc /mnt /cdrom /floppy
Ignore Dirs: /home
Forbidden  : /home /tmp
  (enforcing removal of s[ug]id bits in forbidden paths)

No changes found
```

This checks for changes by recursing the current working directory. Log files will not be rotated and no email sent. All output will go to stdout.

Further documentation

For more details, there are some manual pages you can read:

```
$ man sxid.conf (5)  - Configuration settings for sxid.
$ man sxid (1)       - Check for changes in s[ug]id files and directories.
```

CHAPTER 19

LogSentry

IN THIS CHAPTER

1. Compiling - Optimizing & Installing `LogSentry`
2. Configuring `LogSentry`

Linux LogSentry

Abstract
One of the most important tasks in the security world is to regularly check the log files. Often the daily activities of an administrator doesn't allow them the time to do this task and this can bring about problems.

Don't let the media image fool you, most hackers you'll run across are not very crafty and make a lot of noise ratting your system's door knob…then again they can be as noisy as they want really because there is a 99.99% chance the system administrator won't know anyway <Craig>.

Auditing and logging system events is important! What's more important is that system administrators be aware of these events, so they can prevent problems that will inevitably occur if you have a system connected to the Internet. Unfortunately for most UNIX administrators, it doesn't matter how much you log activity if nobody ever checks the logs, which is often the case. This is where LogSentry also knows in the past as Logcheck will help.

LogSentry automates the auditing process and weeds out "normal" log information to give you a condensed look at problems and potential troublemakers and then mailed to wherever you please. It is a software package that is designed to automatically run and check system log files for security violations and unusual activity by utilizing a program called "logtail" that remembers the last position it read from in a log file and uses this position on subsequent runs to process new information.

These installation instructions assume
Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest LogSentry version number is 1.1.1

The procedures given in this chapter are likely to work on all Linux platforms, but we have only tested it on OpenNA Linux and Red Hat Linux.

Packages
Please check http://www.psionic.com/products/logsentry.html regularly for the latest status. We chose to install from source because it provides the facility to fine tune the installation.

Source code is available from:
LogSentry Homepage: http://www.psionic.com/
You must be sure to download: logsentry-1.1.1.tar.gz

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all the files installed onto the system if you want to update the package in the future. To solve this problem, it's a good idea to make a list of files on the system before you install LogSentry, and then one afterwards, and then compare them using the **diff** utility to find out what files were placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep root]# find /* > LogSentry1
  ```

- And the following one after you install the software:
  ```
  [root@deep root]# find /* > LogSentry2
  ```

- Then use the following command to get a list of what changed:
  ```
  [root@deep root]# diff LogSentry1 LogSentry2 > LogSentry-Installed
  ```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. In our example above, we use the /root directory of the system to store all the generated file lists.

Compiling - Optimizing & Installing LogSentry

Below are the steps that you must make to configure, compile and optimize the LogSentry software before installing it on your system. First off, we install the program as user 'root' so as to avoid any authorization problems.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- This can be done with the following commands:
  ```
  [root@deep /]# cp logsentry-version.tar.gz /var/tmp/
  [root@deep /]# cd /var/tmp/
  [root@deep tmp]# tar xzpf logsentry-version.tar.gz
  ```

Step 2
In order to check that the version of LogSentry, which you are going to install, is an original and unmodified one, use the command described below to check its MD5 hashes checksum.

- To verify the MD5 checksum of LogSentry, use the following command:
  ```
  [root@deep tmp]# md5sum logsentry-1.1.1.tar.gz
  ```

This should yield an output similar to this:
```
e97c2f096e219e20310c1b80e9e1bc29  logsentry-1.1.1.tar.gz
```

Now check that this checksum is exactly the same as the one published on the LogSentry website at the following URL: http://www.psionic.com/downloads/checksums.md5

Step 3
There are some source files to modify before going into the configuration and compilation of the program; the changes allow us to configure the program for our `PATH` environment variable under Linux. Therefore, move into the newly created `LogSentry` source directory and perform the following steps to configure and optimize the software for your system.

- To move into the newly created `LogSentry` directory use the following command:
 `[root@deep tmp]#` **`cd logcheck-1.1.1/`**

Step 4
Here, we have to change default locations of different `LogSentry` configuration files on the system. To archive these modifications, we must edit the `logcheck.sh` script file as follow.

- Edit the **`logcheck.sh`** file and change all of the targeted lines in the order shown below:

a) `vi +47 systems/linux/logcheck.sh` and change the line:

```
LOGTAIL=/usr/local/bin/logtail
```

To read:

`LOGTAIL=/usr/bin/logtail`

b) `vi +55 systems/linux/logcheck.sh` and change the line:

```
TMPDIR=/usr/local/etc/tmp
```

To read:

`TMPDIR=/var/logsentry`

c) `vi +92 systems/linux/logcheck.sh` and change the line:

HACKING_FILE=/usr/local/etc/logcheck.hacking

To read:

HACKING_FILE=/etc/logsentry/hacking

d) `vi +101 systems/linux/logcheck.sh` and change the line:

```
VIOLATIONS_FILE=/usr/local/etc/logcheck.violations
```

To read:

`VIOLATIONS_FILE=/etc/logsentry/violations`

e) `vi +118 systems/linux/logcheck.sh` and change the line:

```
VIOLATIONS_IGNORE_FILE=/usr/local/etc/logcheck.violations.ignore
```

To read:

```
VIOLATIONS_IGNORE_FILE=/etc/logsentry/violations.ignore
```

f) `vi +125 systems/linux/logcheck.sh` and change the line:

```
IGNORE_FILE=/usr/local/etc/logcheck.ignore
```

To read:

```
IGNORE_FILE=/etc/logsentry/ignore
```

Step 5
The `Makefile` file of `LogSentry` needs some modifications too. As for the previous step, we will change default locations of some `LogSentry` files, binary and will add the required optimization `FLAGS` for our `CPU` architecture.

- Edit the **`Makefile`** file and change all of the targeted lines in the order shown below:

a) `vi +14 Makefile` and change the line:

```
CFLAGS = -O
```

To read:

```
CFLAGS = -O2 -march=i686 -funroll-loops
```

b) `vi +22 Makefile` and change the line:

```
INSTALLDIR = /usr/local/etc
```

To read:

```
INSTALLDIR = /etc/logsentry
```

c) `vi +25 Makefile` and change the line:

```
INSTALLDIR_BIN = /usr/local/bin
```

To read:

```
INSTALLDIR_BIN = /usr/bin
```

d) `vi +30 Makefile` and change the line:

```
INSTALLDIR_SH = /usr/local/etc
```

To read:

```
INSTALLDIR_SH = /usr/sbin
```

e) `vi +30 Makefile` and change the line:

```
TMPDIR = /usr/local/etc/tmp
```

To read:

```
TMPDIR = /var/logsentry
```

Step 6
Now, we must make a list of all files on the system before installing the software, and one afterwards, then compare them using the **diff** utility to find out what files are placed where and finally we install the LogSentry software:

```
[root@deep logcheck-1.1.1]# cd
[root@deep root]# find /* > LogSentry1
[root@deep root]# cd /var/tmp/logcheck-1.1.1/
[root@deep logcheck-1.1.1]# mkdir –m0700 /etc/logsentry
[root@deep logcheck-1.1.1]# make linux
[root@deep logcheck-1.1.1]# strip /usr/bin/logtail
[root@deep logcheck-1.1.1]# cd /etc/logsentry/
[root@deep logsentry]# mv logcheck.hacking hacking
[root@deep logsentry]# mv logcheck.violations violations
[root@deep logsentry]# mv logcheck.violations.ignore violations.ignore
[root@deep logsentry]# mv logcheck.ignore ignore
[root@deep logsentry]# cd
[root@deep root]# find /* > LogSentry2
[root@deep root]# diff LogSentry1 LogSentry2 > LogSentry-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

Step 7
Once the configuration, optimization, compilation, and installation of the LogSentry software have been accomplished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete LogSentry and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf logcheck-version/
[root@deep tmp]# rm -f logsentry-version.tar.gz
```

The rm command as used above will remove all the source files we have used to compile and install LogSentry. It will also remove the LogSentry compressed archive from the /var/tmp directory.

Configuring LogSentry

After LogSentry has been built and installed successfully in your system, your next step is to check its configuration files to see if they fit your needs.

- ✓ /etc/logsentry/hacking: (The LogSentry Hacking File)
- ✓ /etc/logsentry/ignore: (The LogSentry Ignore File)
- ✓ /etc/logsentry/violations: (The LogSentry Violation File)
- ✓ /etc/logsentry/violations.ignore: (The LogSentry Violation Ignore File)

From the default install, there are no LogSentry configuration files to modify, the default entries look fine and if you want to make some personal adjustment, all you have to do is to edit the related LogSentry configuration files located under /etc/logsentry directory.

More information about the operation of each one is contained in the INSTALL file of LogSentry under its uncompressed source directory.

Although the fact that there is no LogSentry configuration files to change, the last action to make before using the program is to automate it.

Step 1
Create a file called logsentry under the /etc/cron.daily directory and add the following lines to set LogSentry to run once per day.

- To create the logsentry file under /etc/cron.daily directory, type the following lines in your terminal (as root):

```
cat <<EOF > /etc/cron.daily/logsentry
#!/bin/sh
# Hourly check Log files for security violations and unusual activity.
/usr/bin/logcheck.sh
EOF
```

Step 2
Now, make this script executable and change its permissions to be 0510.

- This can be done with the following commands:
 `[root@deep /]# chmod 510 /etc/cron.daily/logsentry`

WARNING: Remember, in our configuration and installation, LogSentry does not report anything via email if it has nothing useful to say.

CHAPTER 20

HostSentry

IN THIS CHAPTER

1. **Compiling - Optimizing & Installing `HostSentry`**
2. **Configuring `HostSentry`**

Linux HostSentry

Abstract

On Linux servers to accomplish various administrative tasks it is important to have shell access. This shell access can be made from a remote connection or from a local connection but it doesn't matter, we always need to have some shell access on the system and it's rare, if not impossible, to never have the requirement to login in to the server. At least, the super-user "root" will be allowed to get access to the system and for this reason it becomes clear that a tool which can help us to monitor who's connected on the Linux server is important.

Fortunately, a tool exists and it's called "HostSentry" from Psionic Technologies again. HostSentry is a host based intrusion detection tool that performs **L**ogin **A**nomaly **D**etection (LAD). This tool allows administrators to spot strange login behavior and quickly respond to compromised accounts and unusual behavior. We can use it on all servers where shell access is allowed on the system, for known and trusted, users to spot a login problem before it becomes an embarrassing incident.

When HostSentry is installed on your server, a large number of useful possibilities begin to emerge from a single login record and we can track and avoid an anomalous event that seems just a little out of place for a known user.

For example imagine the following:
- Carol the web master, suddenly logs into her shell account at 5:00 AM from China, Sweden, Malaysia and South Korea.

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest HostSentry version number is 0.02

The procedures given in this chapter are likely to work on all Linux platforms, but we have only tested it on OpenNA Linux and Red Hat Linux.

Packages

Please check http://www.psionic.com/products/hostsentry.html regularly for the latest status. We chose to install from source because it provides the facility to fine tune the installation.

Source code is available from:
HostSentry Homepage: http://www.psionic.com/
You must be sure to download: hostsentry-0.02.tar.gz

Prerequisites

HostSentry requires that the listed software below be already installed on your system to be able to compile successfully. If this is not the case, you must install it from your Linux CD-ROM or source archive files. Please make sure you have this program installed on your machine before you proceed with this chapter.

- ✓ Python, which allows HostSentry to run, must already be installed on your system to be able to compile and use the HostSentry software.

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all the files installed onto the system if you want to update the package in the future. To solve this problem, it's a good idea to make a list of files on the system before you install HostSentry, and then one afterwards, and then compare them using the **diff** utility to find out what files were placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep root]# find /* > HostSentry1
  ```
- And the following one after you install the software:
  ```
  [root@deep root]# find /* > HostSentry2
  ```
- Then use the following command to get a list of what changed:
  ```
  [root@deep root]# diff HostSentry1 HostSentry2 > HostSentry-Installed
  ```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. In our example above, we use the /root directory of the system to store all the generated file lists.

Compiling - Optimizing & Installing HostSentry

Below are the steps that you must make to configure, compile and optimize the HostSentry software before installing it on your system. First off, we install the program as user 'root' so as to avoid any authorization problems.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- This can be done with the following commands:
  ```
  [root@deep /]# cp hostsentry-version.tar.gz /var/tmp/
  [root@deep /]# cd /var/tmp/
  [root@deep tmp]# tar xzpf hostsentry-version.tar.gz
  ```

Step 2
In order to check that the version of HostSentry, which you are going to install, is an original and unmodified one, use the command described below to check its MD5 hashes checksum.

- To verify the MD5 checksum of HostSentry, use the following command:
 `[root@deep tmp]# md5sum hostsentry-0.02.tar.gz`

This should yield an output similar to this:
`3de0bbb7d456bb53683de56dfdf98362  hostsentry-0.02.tar.gz`

Now check that this checksum is exactly the same as the one published on the HostSentry website at the following URL: http://www.psionic.com/downloads/checksums.md5

Step 3
There are some source files to modify before going into the configuration and compilation of the program; the changes allow us to configure the program for our PATH environment variable under Linux. Therefore, move into the newly created HostSentry source directory and perform the following steps to configure and optimize the software for your system.

- To move into the newly created HostSentry directory use the following command:
 `[root@deep tmp]# cd hostsentry-0.02/`

Step 4
First, we have to define directories where we want HostSentry to be installed on our system. Editing the Makefile script file as follows does this:

- Edit the **Makefile** file and change all of the targeted lines in the order shown below:

a) `vi +7 Makefile` and change the line:

```
INSTALLDIR = /usr/local/abacus/hostsentry
```

To read:

```
INSTALLDIR = /etc/hostsentry
LIBDIR= /usr/lib/hostsentry
```

b) `vi +21 Makefile` and change the lines:

```
@echo "Installing HostSentry in: $(INSTALLDIR)"
install  -d -g 0 -o root -m 0700 $(INSTALLDIR)
install  -d -g 0 -o root -m 0700 $(INSTALLDIR)/modules
install  -g 0 -o root -m 0700 host* $(INSTALLDIR)
install  -g 0 -o root -m 0700 module* $(INSTALLDIR)/modules
```

To read:

```
install  -d -m 0700 $(INSTALLDIR)
install  -d -m 0700 $(LIBDIR)/modules
install  -m 0400 host* $(LIBDIR)
install  -m 0400 module* $(LIBDIR)/modules
```

Step 5
Once we have defined directories where we want to install the program, we have to change the default locations of some `HostSentry` files, and modules.

- Edit the **`hostSentryConfig.py`** file (`vi +38 hostSentryConfig.py`) and change the line:

  ```
  CONFIG='/usr/local/abacus/hostsentry/hostsentry.conf'
  ```

 To read:

  ```
  CONFIG='/etc/hostsentry/hostsentry.conf'
  ```

- Edit the **`hostSentryStat.py`** file (`vi +141 hostSentryStat.py`) and change the line:

  ```
  db = '/usr/local/abacus/hostsentry/hostsentry.db'
  ```

 To read:

  ```
  db = '/var/hostsentry/hostsentry.db'
  ```

- Edit the **`moduleForeignDomain.py`** file (`vi +45 moduleForeignDomain.py`) and change the line:

  ```
  ALLOW_FILE = '/moduleForeignDomain.allow'
  ```

 To read:

  ```
  ALLOW_FILE = 'moduleForeignDomain.allow'
  ```

- Edit the **`moduleForeignDomain.py`** file (`vi +63 moduleForeignDomain.py`) and change the line:

  ```
  allowPath = config.parseToken('MODULE_PATH')
  ```

 To read:

  ```
  allowPath = '/etc/hostsentry/'
  ```

- Edit the **`moduleMultipleLogins.py`** file (`vi +49 moduleMultipleLogins.py`) and change the line:

  ```
  ALLOW_FILE = '/moduleMultipleLogins.allow'
  ```

 To read:

  ```
  ALLOW_FILE = 'moduleMultipleLogins.allow'
  ```

- Edit the **moduleMultipleLogins.py** file (`vi +78 moduleMultipleLogins.py`) and change the line:

  ```
  allowPath = config.parseToken('MODULE_PATH')
  ```

 To read:

  ```
  allowPath = '/etc/hostsentry/'
  ```

Step 6
Finally, we have to edit the `hostsentry.py` file and add a new line at the BEGINNING of the file to set the environment variable of the `python` binary for the program to find and use it when it runs.

- Edit the **hostsentry.py** file (`vi hostsentry.py`) and add the line:

  ```
  #!/usr/bin/env python
  ```

Step 7
Now, we must make a list of all files on the system before installing the software, and one afterwards, then compare them using the **diff** utility to find out what files are placed where and finally we install the `HostSentry` software:

```
[root@deep hostsentry-0.02]# cd
[root@deep root]# find /* > HostSentry1
[root@deep root]# cd /var/tmp/hostsentry-0.02/
[root@deep hostsentry-0.02]# make install
[root@deep hostsentry-0.02]# mkdir -m0700 /var/hostsentry
[root@deep logsentry]# cd
[root@deep root]# find /* > HostSentry2
[root@deep root]# diff HostSentry1 HostSentry2 > HostSentry-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

Step 8
Once the configuration, optimization, compilation, and installation of the `HostSentry` software have been accomplished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete `HostSentry` and its related source directory, use the following commands:

  ```
  [root@deep /]# cd /var/tmp/
  [root@deep tmp]# rm -rf hostsentry-version/
  [root@deep tmp]# rm -f hostsentry-version.tar.gz
  ```

Configuring HostSentry

After HostSentry has been built and installed successfully in your system, your next step is to configure and customize its configuration files.

- ✓ /etc/hostsentry/hostsentry.conf: (The HostSentry Configuration File)
- ✓ /etc/hostentry/hostsentry.ignore: (The HostSentry Ignore File)
- ✓ /etc/hostentry/hostsentry.action: (The HostSentry Action File)
- ✓ /etc/hostentry/hostsentry.modules: (The HostSentry Modules File)
- ✓ /etc/hostentry/moduleForeignDomain.allow
- ✓ /etc/hostentry/moduleMultipleLogins.allow
- ✓ /etc/init.d/hostsentry: (The HostSentry Initialization File)

/etc/hostsentry/hostsentry.conf: The HostSentry Config File

The hostsentry.conf file is the main configuration file for HostSentry. It is in this file that HostSentry gets all of its information about the way it should run on your system.

Step 1
By default, the hostsentry.conf file does not exist after installation and we have to create it.

- Create the **hostsentry.conf** file (touch /etc/hostsentry/hostsentry.conf) and set your needs. Below is what we recommend you have in your file.

```
IGNORE_FILE = "/etc/hostsentry/hostsentry.ignore"
ACTION_FILE = "/etc/hostsentry/hostsentry.action"
MODULE_FILE = "/etc/hostsentry/hostsentry.modules"
MODULE_PATH = "/usr/lib/hostsentry/modules"
WTMP_FILE = "/var/log/wtmp"
DB_FILE = "/var/hostsentry/hostsentry.db"
DB_TTY_FILE = "/var/hostsentry/hostsentry.tty.db"
WTMP_FORMAT = "384/8:32/44:32/76:256"
```

Step2
Now, set the permissions of the hostsentry.conf file to be (0600/-rw-------) and owned by the super-user 'root' for security reasons.

- To change the permissions and ownership of the hostsentry.conf file, use:

```
[root@deep /]# chmod 600 /etc/hostsentry/hostsentry.conf
[root@deep /]# chown 0.0 /etc/hostsentry/hostsentry.conf
```

/etc/hostsentry/hostsentry.ignore: The HostSentry Ignore File

The hostsentry.ignore file contains a list of users you want HostSentry to never process or never take action against with the modules. This is useful for users such as "ftp" who show up in wtmp but would cause a large number of false alarms because of the anonymous access. It is important to note that each user must be placed one per line.

Step 1
By default, the hostsentry.ignore file doesn't exist after installation and we have to create it.

- Create the **hostsentry.ignore** file (touch /etc/hostsentry/hostsentry.ignore) and put in any user account you want to be ignored if it logs in to the system. Below is what we recommend you put in the file. By default, we have no users defined.

```
# Place usernames in this file that you want to ignore (ftp, etc.)
```

Step2
Now, set the permissions of the hostsentry.ignore file to be (0600/-rw-------) and owned by the super-user 'root' for security reasons.

- To change the permissions and ownership of the hostsentry.ignore file, use:

```
[root@deep /]# chmod 600 /etc/hostsentry/hostsentry.ignore
[root@deep /]# chown 0.0 /etc/hostsentry/hostsentry.ignore
```

/etc/hostsentry/hostsentry.action: The HostSentry Action File

The hostsentry.action file is used to define actions we want HostSentry to take on the specified module. In our example we inform it to logs, blocks the route connection, blocks the TCP connection and disables the user access. It seems that this feature is not implemented but we define and configure it for future version.

Step 1
By default, the hostsentry.action file doesn't exist after installation, so we have to create it manually.

- Create the **hostsentry.action** file (touch /etc/hostsentry/hostsentry.action) and add in any action you want to be taken. Below is what we recommend you enter.

```
moduleFirstLogin=log,blockRoute,blockTCP,disable
```

Step2
Now, set the permissions of the hostsentry.action file to be (0600/-rw-------) and owned by the super-user 'root' for security reasons.

- To change the permission mode and ownership of the hostsentry.action file, use:

```
[root@deep /]# chmod 600 /etc/hostsentry/hostsentry.action
[root@deep /]# chown 0.0 /etc/hostsentry/hostsentry.action
```

/etc/hostsentry/hostsentry.modules: The HostSentry Modules File

The hostsentry.modules file tells HostSentry what modules to execute on login/logout and in what order. If you don't want a particular module to run for whatever reason (false alarms, not interested, etc.) then delete it here.

Step 1
By default, the hostsentry.modules file doesn't exist after installation, so we have to create it.

- Create **hostsentry.modules** file (touch /etc/hostsentry/hostsentry.modules) and add the following module lines to the file. Below is what we recommend.

```
moduleLoginLogout
moduleFirstLogin
moduleForeignDomain
moduleMultipleLogins
moduleRhostsCheck
moduleHistoryTruncated
moduleOddDirnames
```

Step2
Now, set the permissions of the hostsentry.modules file to be (0600/-rw-------) and owned by the super-user 'root' for security reasons.

- To change the permission and ownership of the hostsentry.modules file, use:

```
[root@deep /]# chmod 600 /etc/hostsentry/hostsentry.modules
[root@deep /]# chown 0.0 /etc/hostsentry/hostsentry.modules
```

/etc/hostsentry/moduleForeignDomain.allow

The moduleForeignDomain.allow file is used to list all domains from which we don't want an alert to be sent to the administrator when they log in to the system. Every domain listed in this file will be processed as allowed domains. I recommend you only add your localhost to this file.

Step 1
By default, the moduleForeignDomain.allow file doesn't exist after installation and we have to create it.

- Create the **moduleForeignDomain.allow** file (touch /etc/hostsentry/moduleForeignDomain.allow) and add the following line.

```
:0.0
```

Step2
Now, set the permissions of the moduleForeignDomain.allow file to be (0600/-rw-------) and owned by the super-user 'root' for security reasons.

- To change the permission mode and ownership of the moduleForeignDomain.allow file, use the following commands:

```
[root@deep /]# chmod 600 /etc/hostsentry/moduleForeignDomain.allow
[root@deep /]# chown 0.0 /etc/hostsentry/moduleForeignDomain.allow
```

/etc/hostsentry/moduleMultipleLogins.allow

The moduleMultipleLogins.allow file is used to list all hosts from which multiple loggings are allowed. This mean that all hosts listed in this file will be allowed to make multiple connections from different place without an alert to be send to the administrator. Again, I recommend you to only add your localhost to this file as we do below.

Step 1
By default, the moduleMultipleLogins.allow file does not exist after installation; therefore we have to create it.

- Create the **moduleMultipleLogins.allow** file (touch /etc/hostsentry/moduleMultipleLogins.allow) and add the following line. Below is what we recommend.

```
# Place hosts in here you want this module to disregard logins from.
localhost
```

Step2
Now, set the permissions of the moduleMultipleLogins.allow file to be (0600/-rw-------) and owned by the super-user 'root' for security reasons.

- To change the permission mode and ownership of the moduleMultipleLogins.allow file, use the following commands:

```
[root@deep /]# chmod 600 /etc/hostsentry/moduleMultipleLogins.allow
[root@deep /]# chown 0.0 /etc/hostsentry/moduleMultipleLogins.allow
```

/etc/init.d/hostsentry: The HostSentry Initialization File

The /etc/init.d/hostsentry script file is responsible to automatically starting and stopping the HostSentry server on your Linux system.

Please note that the following script is suitable for Linux operating systems that use SystemV. If you Linux system use some other methods like BSD, you'll have to adjust the script bellow to make it work for you.

Step 1
Create the **hostsentry** script file (touch /etc/init.d/hostsentry) and add the following lines inside it:

```
#!/bin/bash

# This shell script takes care of starting and stopping HostSentry.
#
# chkconfig: 345 98 85
# description: HostSentry is a host based intrusion detection tool that \
#              performs Login Anomaly Detection (LAD). This tool allows \
#              administrators to spot strange login behavior and quickly \
#              respond to compromised accounts and unusual behavior.
#
# processname: hostsentry
# config: /etc/hostsentry/hostsentry.conf
# pidfile: /var/run/hostsentry.pid

# Source function library.
. /etc/init.d/functions
```

```
# Source networking configuration.
. /etc/sysconfig/network

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

RETVAL=0
prog="HostSentry"

start() {
        if [ -f /var/run/hostsentry.pid ] ; then
          pid=`cat /var/run/hostsentry.pid`
          if [ "$pid" != "" ] ; then
              echo $"HostSentry is already running"
              exit 0
          fi
        fi

        echo -n $"Starting $prog: "
        cd /usr/lib/hostsentry
        daemon python hostsentry.py
        RETVAL=$?
        echo
        echo `ps aux | grep "python hostsentry.py" | cut --delimiter=" " -f 7`
>
 /var/run/hostsentry.pid
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/hostsentry
        return $RETVAL
}

stop() {
        echo -n $"Shutting down $prog: "
        cd /usr/lib/hostsentry
        killproc python hostsentry.py
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/hostsentry && rm -f
/var/run
/hostsentry.pid
        return $RETVAL
}

restart() {
        stop
        start
}

condrestart() {
        if [ -f /var/lock/subsys/hostsentry ]; then
          restart
        fi
}

# See how we were called.
case "$1" in
    start)
        start
        ;;
    stop)
        stop
        ;;
    restart)
        restart
```

```
        ;;
    condrestart)
        condrestart
        ;;
    *)
        echo $"Usage: $0 {start|stop|restart|condrestart}"
        exit 1
        ;;
esac
```

Step 2
Once the `/etc/init.d/hostsentry` script file has been created, it is important to make it executable, change its default permissions, create the necessary links and start it. Making this file executable will allow the system to run it, changing its default permission is to allow only the root user to change this file for security reason, and creation of the symbolic links will let your system start the program automatically for you at each system boot.

- To make this script executable and to change its default permissions, use the commands:
  ```
  [root@deep /]# chmod 700 /etc/init.d/hostsentry
  [root@deep /]# chown 0.0 /etc/init.d/hostsentry
  ```

- To create the symbolic `rc.d` links for `HostSentry`, use the following commands:
  ```
  [root@deep /]# chkconfig --add hostsentry
  [root@deep /]# chkconfig --level 345 hostsentry on
  ```

- To start `HostSentry` software manually, use the following command:
  ```
  [root@deep /]# /etc/init.d/hostsentry start
  Starting HostSentry:          [OK]
  ```

CHAPTER 21

PortSentry

IN THIS CHAPTER

1. Compiling - Optimizing & Installing `PortSentry`
2. Configuring `Portsentry`
3. Removing hosts that have been blocked by `PortSentry`

Linux PortSentry

Abstract
Firewalls help us to protect our network from intruders. With them, we can choose which ports we want to open and which ones we don't. This information is kept private by your organization. Nobody on the outside knows this information, but attackers, as well as spammers, know that for some kinds of attacks you can use a special program to scan all the ports on a server to gleam this valuable information (what is open and what is not).

A port scan is a symptom of a larger problem coming your way. It is often the pre-cursor for an attack and is a critical piece of information for properly defending your information resources. PortSentry is a program designed to detect and respond to port scans against a target host in real-time and has a number of options to detect port scans. When it finds one it can react in the following ways:

- ✓ A log indicating the incident is made via syslog().
- ✓ The target host is automatically dropped.
- ✓ The local host is automatically re-configured to route all traffic to the target to a dead host to make the target system disappear.
- ✓ The local host is automatically re-configured to drop all packets from the target via a local packet filter.

The purpose of this is to give to a system administrator a heads up that its host is being probed.

These installation instructions assume
Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest PortSentry version number is 1.1

The procedures given in this chapter are likely to work on all Linux platforms, but we have only tested it on OpenNA Linux and Red Hat Linux.

Packages
Please check http://www.psionic.com/products/portsentry.html regularly for the latest status. We chose to install from source because it provides the facility to fine tune the installation.

Source code is available from:
PortSentry Homepage: http://www.psionic.com/
You must be sure to download: portsentry-1.1.tar.gz

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all the files installed onto the system if you want to update the package in the future. To solve this problem, it's a good idea to make a list of files on the system before you install PortSentry, and then one afterwards, and then compare them using the **diff** utility to find out what files were placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep root]# find /* > PortSentry1
  ```

- And the following one after you install the software:
  ```
  [root@deep root]# find /* > PortSentry2
  ```

- Then use the following command to get a list of what changed:
  ```
  [root@deep root]# diff PortSentry1 PortSentry2 > PortSentry-Installed
  ```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. In our example above, we use the /root directory of the system to store all the generated file lists.

Compiling - Optimizing & Installing PortSentry

Below are the steps that you must make to configure, compile and optimize the PortSentry software before installing it on your system. First off, we install the program as user 'root' so as to avoid any authorization problems.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- This can be done with the following commands:
  ```
  [root@deep /]# cp portsentry-version.tar.gz /var/tmp/
  [root@deep /]# cd /var/tmp/
  [root@deep tmp]# tar xzpf portsentry-version.tar.gz
  ```

Step 2
In order to check that the version of PortSentry, which you are going to install, is an original and unmodified one, use the command described below to check its MD5 hashes checksum.

- To verify the MD5 checksum of PortSentry, use the following command:
  ```
  [root@deep tmp]# md5sum portsentry-1.1.tar.gz
  ```

This should yield an output similar to this:
```
782839446b7eca554bb1880ef0882670  portsentry-1.1.tar.gz
```

Now check that this checksum is exactly the same as the one published on the PortSentry website at the following URL: http://www.psionic.com/downloads/checksums.md5

Step 3
There are some source files to modify before going into the configuration and compilation of the program; the changes allow us to configure the program for our PATH environment variable under Linux. Therefore, move into the newly created PortSentry source directory and perform the following steps to configure and optimize the software for your system.

- To move into the newly created PortSentry directory use the following command:
 [root@deep tmp]# **cd portsentry-1.1/**

Step 4
Here, we have to change default locations of different PortSentry configuration files on the system and add the required optimization FLAGS for our CPU architecture. To make these modifications, we must edit the Makefile script file as follows.

- Edit the **Makefile** file and change all of the targeted lines in the order shown below:

a) vi +29 Makefile and change the line:

```
CFLAGS = -O -Wall
```

To read:

```
CFLAGS = -O2 -march=i686 -funroll-loops -Wall
```

b) vi +40 Makefile and change the lines:

```
INSTALLDIR = /usr/local/psionic
CHILDDIR=/portsentry
```

To read:

```
INSTALLDIR = /usr/sbin
CONFIGDIR = /etc/portsentry
```

c) vi +71 Makefile and change the line:

```
@if [ ! -d $(INSTALLDIR) ]; then /bin/mkdir $(INSTALLDIR); fi
```

To read:

```
@if [ ! -d $(INSTALLDIR) ]; then /bin/mkdir -p $(INSTALLDIR); fi
```

d) vi +73 Makefile and change the lines:

```
@if [ "$(INSTALLDIR)" = "/usr/local/psionic" ]; then /bin/chmod 700 $(INSTALLDIR) ; fi
@echo "Creating portsentry directory $(INSTALLDIR)$(CHILDDIR)"
@if [ ! -d $(INSTALLDIR)$(CHILDDIR) ]; then /bin/mkdir $(INSTALLDIR)$(CHILDDIR); fi
```

To read:

```
@if [ "$(INSTALLDIR)" = "/usr/sbin" ]; then /bin/chmod 700 $(INSTALLDIR) ; fi
@echo "Creating portsentry directory $(CONFIGDIR)"
@if [ ! -d $(CONFIGDIR) ]; then /bin/mkdir -p $(CONFIGDIR); fi
```

e) `vi +77 Makefile` and change the line:

```
chmod 700 $(INSTALLDIR)$(CHILDDIR)
```

To read:

```
chmod 700 $(CONFIGDIR)
```

f) `vi +79 Makefile` and change the lines:

```
cp ./portsentry.conf $(INSTALLDIR)$(CHILDDIR)
cp ./portsentry.ignore $(INSTALLDIR)$(CHILDDIR)
cp ./portsentry $(INSTALLDIR)$(CHILDDIR)
```

To read:

```
cp ./portsentry.conf $(CONFIGDIR)
cp ./portsentry.ignore $(CONFIGDIR)
cp ./portsentry $(INSTALLDIR)
```

g) `vi +83 Makefile` and change the lines:

```
chmod 600 $(INSTALLDIR)$(CHILDDIR)/portsentry.ignore
chmod 600 $(INSTALLDIR)$(CHILDDIR)/portsentry.conf
chmod 700 $(INSTALLDIR)$(CHILDDIR)/portsentry
```

To read:

```
chmod 600 $(CONFIGDIR)/portsentry.ignore
chmod 600 $(CONFIGDIR)/portsentry.conf
chmod 500 $(INSTALLDIR)/portsentry
```

h) `vi +88 Makefile` and change the lines:

```
@echo "Edit $(INSTALLDIR)$(CHILDDIR)/portsentry.conf and change"
```

To read:

```
@echo "Edit $(CONFIGDIR)/portsentry.conf and change"
```

i) `vi +94 Makefile` and change the lines:

```
@echo "and config files ($(INSTALLDIR)$(CHILDDIR))."
```

To read:

```
@echo "and config files $(CONFIGDIR)."
```

Step 5
The second file that we will modify is the `portsentry_config.h` header file. In this file, we will change the default install location of the configuration file for `PortSentry`.

- Edit the **`portsentry_config.h`** file (`vi +32 portsentry_config.h`) and change the following line:

  ```
  #define CONFIG_FILE "/usr/local/psionic/portsentry/portsentry.conf"
  ```

 To read:

  ```
  #define CONFIG_FILE "/etc/portsentry/portsentry.conf"
  ```

Step 6
Now, we must make a list of all files on the system before installing the software, and one afterwards, then compare them using the **`diff`** utility to find out what files are placed where and finally we install the `PortSentry` software:

```
[root@deep portsentry-1.1]# cd
[root@deep root]# find /* > PortSentry1
[root@deep root]# cd /var/tmp/portsentry-1.1/
[root@deep portsentry-1.1]# make linux
[root@deep portsentry-1.1]# make install
[root@deep portsentry-1.1]# strip /usr/sbin/portsentry
[root@deep portsentry-1.1]# mkdir -m0700 /var/portsentry
[root@deep portsentry-1.1]# cd
[root@deep root]# find /* > PortSentry2
[root@deep root]# diff PortSentry1 PortSentry2 > PortSentry-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

Step 7
Once the configuration, optimization, compilation, and installation of the `PortSentry` software have been accomplished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete `PortSentry` and its related source directory, use the following commands:

  ```
  [root@deep /]# cd /var/tmp/
  [root@deep tmp]# rm -rf portsentry-version/
  [root@deep tmp]# rm -f portsentry-version.tar.gz
  ```

Configuring `PortSentry`

After `PortSentry` has been built and installed successfully on your system, your next step is to configure and customize its configuration files to fit your needs.

- ✓ `/etc/portsentry/portsentry.conf`: (The `PortSentry` Configuration File)
- ✓ `/etc/portsentry/portsentry.ignore`: (The `PortSentry` Ignore File)
- ✓ `/etc/portsentry/portsentry.modes`: (The `PortSentry` Modes File)
- ✓ `/etc/init.d/portsentry`: (The `PortSentry` Initialization File)

/etc/portsentry/portsentry.conf: The PortSentry Config File

The portsentry.conf file is the main configuration file for PortSentry. It is in this file that PortSentry gets all of its information about the way it should runs your system. You can specify which ports you want PortSentry to listen to, which IP addresses are to be denied, monitored, ignored, or have their automatic responses disabled, and so on.

- Edit the **portsentry.conf** file (vi /etc/portsentry/portsentry.conf) and set your needs. Below is what we recommend.

```
TCP_PORTS="1,11,81,82,83,1080,1720,1863,5190,8080"
UDP_PORTS="1,7,9,81,82,83,1080,1720,1863,5190,8080"
ADVANCED_PORTS_TCP="1024"
ADVANCED_PORTS_UDP="1024"
ADVANCED_EXCLUDE_TCP="113,139"
ADVANCED_EXCLUDE_UDP="520,138,137,67"
IGNORE_FILE="/etc/portsentry/portsentry.ignore"
HISTORY_FILE="/var/portsentry/portsentry.history"
BLOCKED_FILE="/var/portsentry/portsentry.blocked"
RESOLVE_HOST="0"
BLOCK_UDP="0"
BLOCK_TCP="1"
KILL_ROUTE="/sbin/route add -host $TARGET$ reject"
SCAN_TRIGGER="0"
PORT_BANNER="** UNAUTHORIZED ACCESS PROHIBITED **"
```

This tells the posrtsentry.conf file to set itself up for this particular configuration with:

TCP_PORTS="1,11,81,82,83,1080,1720,1863,5190,8080"
The option "TCP_PORTS" specifies which TCP ports we want PortSentry to listen to for scan attacks. It is important to note that this option is used by all PortSentry modes except the Advanced Stealth Scan Detection mode that completely ignores this option because it uses a more advanced and a more secure method to monitor ports. Remember that the Advanced Stealth Scan Detection is what we use in this configuration; therefore we don't really need to define this option. With the other scan detection modes; you have to define all the TCP ports from which you want PortSentry to monitor here.

UDP_PORTS="1,7,9,81,82,83,1080,1720,1863,5190,8080"
This option "UDP_PORTS" specifies which UTP ports we want PortSentry to listen for scan attacks on. As with the above option, it is important to note that this option is used by all PortSentry modes except the Advanced Stealth Scan Detection mode which completely ignores this option because it uses a more advanced and a more secure method to monitor ports. Again, Advanced Stealth Scan Detection is what we use in this configuration; therefore we don't really need to define this option. On other scan detection modes, you have to define here all the UDP ports from which you want PortSentry to monitor.

ADVANCED_PORTS_TCP="1024"
The option "ADVANCED_PORTS_TCP" specifies the highest TCP port number to monitor down from. Any port *below* this number is then monitored by PortSentry in all detection modes. The default is 1024 (reserved port range), and the one we use here for TCP.

ADVANCED_PORTS_UDP="1024"
The option "ADVANCED_PORTS_UDP" specifies the highest UDP port number to monitor down from. Any port *below* this number is then monitored by PortSentry in all detection modes. The default is 1024 (reserved port range), and the one we use here for UDP.

ADVANCED_EXCLUDE_TCP="113,139"
The option "ADVANCED_EXCLUDE_TCP" specifies the TCP ports that should be manually excluded from monitoring in advanced mode. These are normally ports that may get hit by mistake by remote clients and shouldn't cause alarms. The above TCP ports should be ok for most of us.

ADVANCED_EXCLUDE_UDP="520,138,137,67"
The option "ADVANCED_EXCLUDE_UDP" specifies the UDP ports that should be manually excluded from monitoring in advanced mode. Again, these are normally ports that may get hit by mistake by remote clients and shouldn't cause alarms. The above UDP ports should be ok for most of us.

IGNORE_FILE="/etc/portsentry/portsentry.ignore"
The option "IGNORE_FILE" specifies the path to the file that contains IP addresses of hosts you want to always be ignored by PortSentry. See later in this chapter for more information about his file.

HISTORY_FILE="/var/portsentry/portsentry.history"
The option "HISTORY_FILE" specifies the path to the file that contains hosts that have been denied by PortSentry.

BLOCKED_FILE="/var/portsentry/portsentry.blocked"
The option "BLOCKED_FILE" specifies the path to the file that contains the IP addresses of blocked hosts by PortSentry. It is important to note that all IP addresses listed in this file are blocked by PortSentry until the program restarts.

RESOLVE_HOST="0"
The option "RESOLVE_HOST" specifies if we want PortSentry to make DNS resolution or not. In our configuration, we turn off DNS resolution for better performance. The number "1" enable the option and number "0" disable it. This is a performance feature.

BLOCK_UDP="0"
The option "BLOCK_UDP" is used to disable all automatic responses to UDP probes. Because UDP can be easily forged, it may allow an attacker to start a denial of service attack against the protected host, causing it to block all manner of hosts that should normally be left alone. Setting this option to "0" will disable all responses, although the connections are still logged.

BLOCK_TCP="1"
The option "BLOCK_TCP" is the same as above, but for TCP. Packet forgery is not as big a problem though, because PortSentry waits for a full connect to occur and this is much harder to forge in the basic modes. Leave this enabled, even for Internet connected hosts.

KILL_ROUTE="/sbin/route add -host $TARGET$ reject"
The option "KILL_ROUTE" specifies the command to run to drop the offending route if an attack is detected. We can use IPTables here as the command to block attack but it is better to go with the "route" command as we do because IPTables will block the attacker only when connection is closed by the remote host where the "route" command will directly block the attacker. Therefore the above option is more effective and secure than using IPTables command.

SCAN_TRIGGER="0"
PortSentry has a state engine that will remember hosts that connected to it. Setting this value will tell PortSentry to allow X number of grace port hits before it reacts. This will detect both sequential and random port sweeps. The default is 0, which will react immediately.

```
PORT_BANNER="** UNAUTHORIZED ACCESS PROHIBITED **"
```

The option "PORT_BANNER" specifies a text banner you want displayed to the connecting host if the PortSentry is activated.

/etc/portsentry/portsentry.ignore: The PortSentry Ignore File

The portsentry.ignore file is where you add any host you wanted to be ignored if it connects to a tripwired port. This should always contain at least the localhost (127.0.0.1) and the IP's of the local interfaces (lo). It is not recommend that you put in every IP on your network. It is well commented and very simple to understand.

- Edit the **portsentry.ignore** file (vi /etc/portsentry/portsentry.ignore) and add in any host you want to be ignored if it connects to a tripwired port. Below is what we recommend.

```
# Put hosts in here you never want blocked. This includes the IP
# addresses of all local interfaces on the protected host
# (i.e virtual host, mult-home). Keep 127.0.0.1 and 0.0.0.0 to keep
# people from playing games.
#
# PortSentry can support full netmasks for networks as well. Format is:
#
# <IP Address>/<Netmask>
#
# Example:
#
# 192.168.2.0/24
# 192.168.0.0/16
# 192.168.2.1/32
# Etc.
#
# If you don't supply a netmask it is assumed to be 32 bits.
#
#

127.0.0.1/32
0.0.0.0
```

NOTE: Don't forget to add the IP address of your server to the above list. For example, if I've installed PortSentry on one of my server, which has IP address of 207.35.78.3, then I'll add this IP to the above list.

/etc/portsentry/portsentry.modes: The PortSentry Modes File

The PortSentry program can be configured in six different modes of operation but be aware that only one protocol mode type can be started at a time. To be more accurate, you can start one TCP mode and one UDP mode, so two TCP modes and one UDP mode, for example, won't work.

- The available PortSentry modes are:

- ✓ portsentry –tcp (Basic port-bound TCP mode)
- ✓ portsentry –udp (Basic port-bound UDP mode)
- ✓ portsentry –stcp (Stealth TCP scan detection mode)
- ✓ portsentry –sudp ("Stealth" UDP scan detection mode)
- ✓ portsentry –atcp (Advanced "Stealth" TCP scan detection mode)
- ✓ portsentry –audp (Advanced "Stealth" UDP scan detection mode)

For the best use of this software it is preferable to start PortSentry in **Advanced TCP stealth scan detection mode** and **Advanced UDP stealth scan detection mode**. For information about the other modes available, please refer to the README.install and README.stealth file under the PortSentry source directory.

With the **Advanced TCP stealth scan detection mode** "–atcp", PortSentry will first check to see what ports you have running on your server, then remove these ports from monitoring and will begin watching the remaining ports. This is very powerful and reacts exceedingly quickly for port scanners. It also uses very little CPU time. This mode is the most sensitive and the most effective of all the protection options.

The six different modes of operation under which PortSentry can operate must be specified in the configuration file named portsentry.modes located in the /etc/portsentry/ directory. We can add inside this file all the six possible modes of PortSentry, and then uncomment the two we want to use for our server.

Step 1
By default, the portsentry.modes file does not exist after installation, and we have to create it.

- Create the **portsentry.modes** file (touch /etc/portsentry/portsentry.modes) and add the following lines inside the file. Below is what we recommend.

```
# These are the available startup modes for PortSentry. Uncomment the
# modes you want PortSentry to run in. For information about each
# available mode, please see the PortSentry documentation.
#
# Normal TCP/UDP scanning:
#tcp
#udp
#
# Steal TCP/UDP scanning:
#stcp
#sudp
#
# Advanced Stealth TCP/UDP scanning:
atcp
audp
```

Step2
Now, set the permissions of the `portsentry.modes` file to be (`0600/-rw-------`) and owned by the super-user '`root`' for security reasons.

- To change the permission mode and ownership of the `portsentry.modes` file, use:
```
[root@deep /]# chmod 600 /etc/portsentry/portsentry.modes
[root@deep /]# chown 0.0 /etc/portsentry/portsentry.modes
```

`/etc/init.d/portsentry`: The `PortSentry` Initialization File

The `/etc/init.d/portsentry` script file is responsible to automatically starting and stopping the `PortSentry` server on your Linux system.

Please note that the following script is suitable for Linux operating systems that use `SystemV`. If you Linux system use some other methods like `BSD`, you'll have to adjust the script bellow to make it work for you.

Step 1
Create the **`portsentry`** script file (`touch /etc/init.d/portsentry`) and add the following lines inside it:

```
#!/bin/bash

# This shell script takes care of starting and stopping the Port Scan Detector.
#
# chkconfig: 345 98 05
# description: PortSentry Port Scan Detector is part of the Abacus Project \
#              suite of tools. The Abacus Project is an initiative to release \
#              low-maintenance, generic, and reliable host based intrusion \
#              detection software to the Internet community.
#
# processname: portsentry
# config: /etc/portsentry/portsentry.conf
# pidfile: /var/run/portsentry.pid

# Source function library.
. /etc/rc.d/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

RETVAL=0
prog="PortSentry"

start() {
        SENTRYDIR=/etc/portsentry
        if [ -s $SENTRYDIR/portsentry.modes ] ; then
        modes=`cut -d "#" -f 1 $SENTRYDIR/portsentry.modes`
        else
        modes="tcp udp"
        fi

        for i in $modes ; do
        action "Starting $prog -$i: " /usr/sbin/portsentry -$i
        RETVAL=$?
```

```
        done

        echo
        [ $RETVAL = 0 ] && touch /var/lock/subsys/portsentry
        return $RETVAL
}

stop() {
        echo -n "Shutting down $prog: "
        killproc portsentry
        RETVAL=$?
        echo
        [ $RETVAL = 0 ] && rm -f /var/lock/subsys/portsentry
        return $RETVAL
}

# See how we were called.
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  restart|reload)
        stop
        start
        RETVAL=$?
        ;;
  condrestart)
        [ -f /var/lock/subsys/portsentry ] && restart || :
        ;;
  status)
        status portsentry
        ;;
  *)
    echo "Usage: portsentry {start|stop|restart|reload|condrestart|status}"
    exit 1
esac
```

Step 2

Once the `/etc/init.d/portsentry` script file has been created, it is important to make it executable, change its default permissions, create the necessary links and start it. Making this file executable will allow the system to run it, changing its default permission is to allow only the root user to change this file for security reasons, and creation of the symbolic links to start the program automatically for you at each system boot.

- To make this script executable and to change its default permissions, use the command:
  ```
  [root@deep /]# chmod 700 /etc/init.d/portsentry
  [root@deep /]# chown 0.0 /etc/init.d/portsentry
  ```

- To create the symbolic `rc.d` links for `PortSentry`, use the following command:
  ```
  [root@deep /]# chkconfig --add portsentry
  [root@deep /]# chkconfig --level 345 portsentry on
  ```

- To start `PortSentry` software manually, use the following command:
  ```
  [root@deep /]# /etc/init.d/portsentry start
  Starting PortSentry:          [OK]
  ```

Removing hosts that have been blocked by PortSentry

When PortSentry want to blocks attackers it uses the "route" command of Linux to do it. Every host that has been blocked by PortSentry will appear under the 'route" command. Below, I show you how to remove hosts that have been blocked by PortSentry from your system.

Step 1
We have to use the "route" command to list which hosts are presently blocked by the program. The "route" command also lists other important information about your network routing but we use it in this example to get the list of blocked hosts and to unlock them from the system.

- To list which hosts are presently blocked by PortSentry, use the command:

```
[root@deep /]# route
Kernel IP routing table
Destination   Gateway     Genmask         Flags   Metric  Ref Use Iface
www.hack.com  -           255.255.255.255 !H      0       -   0   -
207.35.78.0   *           255.255.255.224 U       0       0   0   eth0
127.0.0.0     *           255.0.0.0       U       0       0   0   lo
default       rt.openna.c 0.0.0.0         UG      0       0   0   eth0
```

In the above example, we can see that "www.hack.com" is listed into our routing table as a domain that has been blocked by PortSentry because it tried to scan our system. The "-" string inform us about the fact that this host is locked. Every host in the routing table with this string "-" is marked as blocked by the system.

Step2
Now that we know about the host that has been blocked by PortSentry, we can decide to remove it from the list of blocked hosts on our system.

- To remove the blocked host in question, use the following command:

```
[root@deep /]# route del -host www.hack.com reject
```

The above command will remove www.hack.com from the list of blocked hosts in the routing table of our system. The option "del" in the "route" command is what makes it possible to remove the host from the list. Your have to use the above command for any additional hosts that you want to remove from the routing table.

Step 3
Finally, we have to edit the portsentry.history file and remove the line corresponding to www.hack.com from the file. This is important for PortSentry to be able to add the site into the list of blocked host in the event that the corresponding host tries to scan your system again.

- Edit the portsentry.history file and remove the line corresponding to the host:

```
[root@deep /]# vi /var/portsentry/portsentry.history
1020371099 - 05/02/2002 16:24:59 Host: 1.2.3.4/1.2.3.4 Port: 80 TCP Blocked
```

CHAPTER 22

Snort

IN THIS CHAPTER

1. **Compiling - Optimizing & Installing `Snort`**
2. **Configuring `Snort`**
3. **Running `Snort` in a chroot jail**

Linux Snort

Abstract
From the point of view of security, information is vital and we have to get as much information as we can to quickly discover problem and possible attack on our network. In previous chapters, we have already installed many useful security programs to help us gather information and stop attacks but this is not enough and we have to add to our arsenal another security tool which can scan our network and report possible problems and attacks. This is where Snort will help us.

Snort is a flexible libpcap-based packet sniffer/logger tool, which can be used in the most classic sense as a lightweight **n**etwork **i**ntrusion **d**etection **s**ystem (NIDS) but it is also useful for a wide variety of other uses. It features rules based logging and can perform protocol analysis, content searching/matching and can be used to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts, and much more. Snort has a real-time alerting capability, with alerts being sent to syslog, a separate "alert" file, or as a WinPopup message via Samba's smbclient.

Network **i**ntrusion **d**etection **s**ystems (NIDS) are an important part of any network security architecture. They provide a layer of defense, which monitors network traffic for predefined suspicious activity or patterns, and alert system administrators when potential hostile traffic is detected. This is exactly what we are looking for here, a lightweight network intrusion detection tool that can be deployed to monitor TCP/IP networks and detect a wide variety of suspicious network traffic as well as outright attacks and can provide administrators with enough data to make informed decisions on the proper course of action in the face of suspicious activity.

Some could say that PortSentry, which we have installed previously, does the same thing. This is NOT true; PortSentry can be used to block unauthorized ports that have been scanned by attackers and nothing else. Snort goes more deeply with the TCP/IP protocol and provides myriad of security information related to many services running on your server. In general, it is a very good security tool to use with all other security tools as discussed on this book. I highly recommend you to install it on your system if you want to be informed about hostile activities and also methods used by spammers, crackers, etc to probe your network.

These installation instructions assume
Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account “root”.
Whether kernel recompilation may be required: No
Latest Snort version number is 1.8.7

The procedures given in this chapter are likely to work on all Linux platforms, but we have only tested it on OpenNA Linux and Red Hat Linux.

Packages

The following is based on information listed by Snort as of 2002/07/08. Please check http://www.snort.org/ regularly for the latest status. We chose to install from source because it provides the facility to fine tune the installation.

Source code is available from:
Snort Homepage: http://www.snort.org/
You must be sure to download: snort-1.8.7.tar.gz

Prerequisites

Snort requires that the listed software below be already installed on your system to be able to compile successfully. If this is not the case, you must install it from your Linux CD-ROM or source archive files. Please make sure you have this program installed on your machine before you proceed with this chapter.

- ✓ Libpcap, which is used extensively by Snort, must already be installed on your system.
- ✓ Tcpdump, which allows some additional features with Snort.

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all the files installed onto the system if you want to update the package in the future. To solve this problem, it's a good idea to make a list of files on the system before you install Snort, and then one afterwards, and then compare them using the **diff** utility to find out what files were placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep root]# find /* > Snort1
  ```
- And the following one after you install the software:
  ```
  [root@deep root]# find /* > Snort2
  ```
- Then use the following command to get a list of what changed:
  ```
  [root@deep root]# diff Snort1 Snort2 > Snort-Installed
  ```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. In our example above, we use the /root directory of the system to store all the generated file lists.

Compiling - Optimizing & Installing Snort

Below are the steps that you must make to configure, compile and optimize the Snort software before installing it on your system. First off, we install the program as user 'root' so as to avoid any authorization problems.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- This can be done with the following commands:

```
[root@deep /]# cp snort-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf snort-version.tar.gz
```

Step 2
In order to check that the version of Snort, which you are going to install, is an original and unmodified one, use the command described below to check its MD5 hashes checksum.

- To verify the MD5 checksum of Snort, use the following command:

```
[root@deep tmp]# md5sum snort-1.8.7.tar.gz
```

This should yield an output similar to this:

```
29c81d0bc243edb21ba4ab33ee80457e  snort-1.8.7.tar.gz
```

Now check that this checksum is exactly the same as the one published on the Snort website at the following URL: http://www.snort.org/dl/snort-1.8.7.tar.gz.md5

Step 3
Snort needs a UID and GID to properly run on the system but this UID/GID cannot run as super-user root; therefore we must create a special user with no shell privileges on the system for running Snort daemon.

- To create this special Snort user on OpenNA Linux, use the following command:

```
[root@deep tmp]# groupadd -g 69 snort > /dev/null 2>&1 || :
[root@deep tmp]# useradd -c "Snort NIDS" -d /var/log/snort -g 69 -s
/bin/false -u 69 snort > /dev/null 2>&1 || :
```

- To create this special Snort user on Red Hat Linux, use the following command:

```
[root@deep tmp]# groupadd -g 69 snort > /dev/null 2>&1 || :
[root@deep tmp]# useradd -u 69 -g 69 -s /bin/false -M -r -d
/var/log/snort snort > /dev/null 2>&1 || :
```

The above command will create a null account, with no password, no valid shell, no files owned-nothing but a UID and a GID for the program. Remember that Snort daemon does not need to have a shell account on the server.

Step 4
Now, edit the **shells** file (`vi /etc/shells`) and add a non-existent shell name "`/bin/false`", which is the one we used in the `passwd` command above.

```
[root@deep tmp]# vi /etc/shells
/bin/bash2
/bin/bash
/bin/sh
/bin/false ← This is our added no-existent shell
```

Step 5
Next, move into the newly created `Snort` source directory and perform the following steps to configure and optimize the software for your system.

- To move into the newly created `Snort` directory use the following command:
 `[root@deep tmp]# cd snort-1.8.7/`

- To configure and optimize `Snort` use the following compilation lines:

```
CFLAGS="-O2 -march=i686 -funroll-loops"; export CFLAGS
./configure \
--prefix=/usr \
--sysconfdir=/etc \
--localstatedir=/var \
--mandir=/usr/share/man \
--with-openssl
```

Step 6
Now, we must make a list of all files on the system before installing the software, and one afterwards, then compare them using the **diff** utility to find out what files are placed where and finally we install the `Snort` software:

```
[root@deep snort-1.8.7]# make
[root@deep snort-1.8.7]# cd
[root@deep root]# find /* > Snort1
[root@deep root]# cd /var/tmp/snort-1.8.7/
[root@deep snort-1.8.7]# make install
[root@deep snort-1.8.7]# mkdir -p /var/log/snort
[root@deep snort-1.8.7]# mkdir -p /etc/snort
[root@deep snort-1.8.7]# chown -R snort.snort /var/log/snort/
[root@deep snort-1.8.7]# install classification.config /etc/snort/
[root@deep snort-1.8.7]# install snort.conf *.rules /etc/snort/
[root@deep snort-1.8.7]# chmod 0644 /etc/snort/*
[root@deep snort-1.8.7]# strip /usr/bin/snort
[root@deep snort-1.8.7]# cd
[root@deep root]# find /* > Snort2
[root@deep root]# diff Snort1 Snort2 > Snort-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

Step 7
Once the configuration, optimization, compilation, and installation of the Snort software have been accomplished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete Snort and its related source directory, use the following commands:
```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf snort-version/
[root@deep tmp]# rm -f snort-version.tar.gz
```

Configuring Snort

After Snort has been built and installed successfully in your system, your next step is to configure and customize its configuration files to fit your needs.

- ✓ /etc/snort/snort.conf: (The Snort Configuration File)
- ✓ /etc/init.d/snort: (The Snort Initialization File)

/etc/snort/snort.conf: The Snort Config File

The snort.conf file is the main configuration file for Snort. It is in this file that Snort gets all of its startup information and the way it should run on your system. You can edit it to specify your network variables, preprocessors parameters, output plugging to use and Snort rules files.

The Snort configuration file is divided into four different sections. The first section is used to define network variables, the second section is used to configure the preprocessor parameters that Snort should use, the third section is used to configure output plugging to activate, and the last section is used to enable specific Snort rule set. Below, we will explain each section and how you should use them to configure Snort for your server.

- Edit the **snort.conf** file (vi /etc/snort/snort.conf) and set your needs. Below is what we recommend you.

```
var HOME_NET $eth0_ADDRESS
var EXTERNAL_NET any
var SMTP $HOME_NET
var HTTP_SERVERS $HOME_NET
var SQL_SERVERS $HOME_NET
var DNS_SERVERS $HOME_NET
var RULE_PATH ./
preprocessor frag2
preprocessor stream4: detect_scans,detect_state_problems
preprocessor stream4_reassemble: both,ports all
preprocessor http_decode: 80
preprocessor rpc_decode: 111 32771
preprocessor bo
preprocessor telnet_decode
preprocessor portscan: $HOME_NET 4 3 portscan.log
preprocessor portscan-ignorehosts: 207.35.78.40 207.35.78.41
output alert_syslog: LOG_AUTH LOG_ALERT
include classification.config
```

This tells the snort.conf file to set itself up for this particular configuration with:

The network variables section

The first section of the Snort configuration file refers to all network parameters specific to your networking architecture and IP information. This section is used by Snort to get information about the way it should monitor your network. Other options are available as shown below.

```
var HOME_NET $eth0_ADDRESS
```

The option "HOME_NET" is used to specify the network interface on which you want Snort to run and listen. The above parameter allows us to initialize Snort for the IP address and netmask of the network interface which we run Snorton. This is very useful for dialup users and even for static IP addresses.

```
var EXTERNAL_NET any
```

The option "EXTERNAL_NET" is used to specify the external network addresses that Snort should monitor. Here we keep the default setting to monitor any external network addresses meaning that if any IP addresses/hosts try to do something with our server, we will know it because Snort will monitor any external network addresses trying to connect, scan, attack, etc our server.

```
var SMTP $HOME_NET
```

The option "SMTP" is used to specify all SMTP servers that Snort should monitor. The default setting is correct for most of us. Here, we simply inform Snort to monitor SMTP servers running on our network. The variable "$HOME_NET" redirects Snort to the network interface and IP address of the server where it runs meaning that if SMTP services are running on our network, Snort will monitor any connection to them.

```
var HTTP_SERVERS $HOME_NET
```

The option "HTTP_SERVERS" is used to specify all HTTP servers that Snort should monitor. Here again, the default setting is good enough for most of us. We simply inform Snort to monitor HTTP servers running on our network. The variable "$HOME_NET" redirects Snort to the network interface and IP address of the server where it runs, meaning that if HTTP services are running on our network, Snort will monitor any connection to them.

```
var SQL_SERVERS $HOME_NET
```

The option "SQL_SERVERS" is used to specify all SQL servers that Snort should monitor. Again, the default setting is the right one for most of us. We inform Snort to monitor SQL servers running on our network. The variable "$HOME_NET" redirects Snort to the network interface and IP address of the server where it runs meaning that if SQL services are running on our network, Snort will monitor any connection to them.

```
var DNS_SERVERS $HOME_NET
```

The option "DNS_SERVERS" is used to specify all DNS servers that Snort should monitor. Again, the default setting is good for most of us. We simply inform Snort to monitor DNS servers running on our network. The variable "$HOME_NET" redirects Snort to the network interface and IP address of the server where it runs meaning that if DNS services are running on our network, Snort will monitor any connection to them.

```
var RULE_PATH ./
```

The option "RULE_PATH" simply specifies the path where all Snort rules files are located on the system. You don't need to change the default setting. Snort use many rules files to get information about what actions to take when an attack is detected. Rule files handle signatures, etc about the specified service. More information about Snort rules can be found later in this chapter.

The preprocessors section
This section of the Snort configuration file is used to define general configuration for preprocessors. Preprocessors are used to define parameters and options that Snort should use when running.

preprocessor frag2
The preprocessor "frag2" enables support for IP defragmentation and fragmentation attacks with Snort. This plug-in will allow Snort to perform IP defragmentation and detect people launching fragmentation attacks (usually DoS) against hosts. The preprocessor has two options associated with it.

The options are "timeout" and "memcap". The "timeout" option could be used to change the default number of seconds an unfinished fragment will be kept around waiting for completion. The second option "memcap" could be used to limit memory usage of IP defragmentation. The default value for both options are correct and we don't need to change them. This is a security feature.

preprocessor stream4: detect_scans,detect_state_problems
The preprocessor "stream4" enables support for full TCP stream reassembly, stateful inspection of TCP streams, etc with Snort. This plug-in will allow Snort to statefully detect various types of portscan, fingerprinting, ECN, etc and will help to defeat stick/snot against TCP rules. This preprocessor has seven options associated with it.

The options are "detect_scans", "detect_state_problems", "keepstats", "noinspect", "timeout", "memcap", and "log_flushed_streams".

- detect_scans: Used to detect stealth portscans and generate alerts.
- detect_state_problems: Used to detect TCP state problems.
- keepstats: Used to keep session statistics.
- noinspect: Used to turn off stateful inspection only.
- timeout: Used to set or change the default session timeout counter.
- memcap: Used to limit memory usage by changing default setting.
- log_flushed_streams: Used to cause all packets that are stored in the packet buffers to be flushed to disk.

In our configuration of this preprocessor, we use "detect_scans" to detect stealth portscans and generate alerts and "detect_state_problems" to detect possible TCP state problems. We don't need the other options. This is a security feature.

preprocessor stream4_reassemble: both,ports all
The preprocessor "stream4_reassemble" is a continuation of the above preprocessor parameter and specifies the tcp stream reassembly directive to use with Snort. This preprocessor has five options associated with it.

The options are "clientonly", "serveronly", "both", "noalerts", and "ports".

- clientonly: Used to reassemble traffic for the client side of a connection only.
- serveronly: Used to reassemble traffic for the server side of a connection only.
- both: Used to reassemble both sides of a session.
- noalerts: Used to turn off alerts from the stream reassembly stage.
- ports: Used to specify the ports number to use for reassembly.

In our configuration of this preprocessor, we use "both" to reassemble both sides of a session, and "ports all" to turn on reassembly for all ports. We don't need the other options. This is a security feature.

preprocessor http_decode: 80
The preprocessor "http_decode" enables support for normalized HTTP requests with Snort. This preprocessor allow us to defeat hostile attackers trying to stealth themselves from IDSs by mixing these substitutions in with the HTTP request.

It has three arguments that you can associate with it. The first is the port number you want it to analyze; this argument should always be present with this preprocessor. The second argument is "-unicode" and you can use it to turn off detection of UNICODE directory traversal attacks. By default, this argument (-unicode) is set with Snort and we remove it in our configuration to make the preprocessor use it.

The last argument "-cginull" related to detection of CGI NULL code attacks with the HTTP protocol. If you add "-cginull" to this preprocessor parameter, you will turn off detection of CGI NULL code attacks. In our configuration we don't specify this argument (-cginull) because we want to use this feature and let Snort detect all possible CGI NULL code attacks on the server.

- 80: Used to specify the port number you want the preprocessor to analyze.
- -unicode: Used to turn off detection of UNICODE directory traversal attacks.
- -cginull: Used to turn off detection of CGI NULL code attacks.

In our configuration with this preprocessor, we only specify the port numbers (80) we want the preprocessor to analyze for HTTP services. We don't need the other arguments. This is a security feature.

preprocessor rpc_decode: 111 32771
The preprocessor "rpc_decode" enables support for normalized RPC traffic in much the same way as the http_decode preprocessor. This plug-in takes the port numbers that RPC services are running on as arguments. In our configuration of this preprocessor, we define ports 111 and 32771. You don't need to change the default setting.

preprocessor bo
The preprocessor "bo" is used to detect **B**ack **O**rifice (bo) traffic on the network. It uses the Back Orifice "encryption" algorithm to search for traffic conforming to the Back Orifice protocol. It provides two arguments that you can associate with it.

The first is "-nobrute" which turns off the plugin's brute forcing routine and the second argument is a number to use as the default key when trying to decrypt the traffic.

- -nobrute: Used to turns off the plugin's brute forcing routine.
- 31337: Used as the default key when trying to decrypt the traffic.

In our configuration of this preprocessor, we use the default setting and don't specify any additional arguments. This is a performance feature.

preprocessor telnet_decode
The preprocessor "telnet_decode" enables support to normalize telnet negotiation strings from telnet and ftp traffic with Snort. It works in much the same way as the http_decode preprocessor, searching for traffic that breaks up the normal data stream of a protocol and replacing it with a normalized representation of that traffic. This preprocessor requires no arguments.

```
preprocessor portscan: $HOME_NET 4 3 portscan.log
```
The preprocessor "portscan" enables support to detect UDP packets or TCP SYN packets going to four different ports in less than three seconds. In our configuration, we log all reports to the "portscan.log" file locate under the /var/log/snort directory.

```
preprocessor portscan-ignorehosts: 207.35.78.40 207.35.78.41
```
The preprocessor "portscan-ignorehosts" is used to list particular host(s) from which we want Snort to ignore traffic. This mean that any host(s) IP address(es) listed in this line will be ignored by Snort. Values are defined in a white space delimited list.

The output plug-in section

This section of the Snort configuration file is used to configure the output plug-in you decide to use with Snort. Snort can be configured to log all reports to an SQL database of your choice or to run with an external security program or even to log all reports to a local file on the system and many other features. In general, we only need to specify some options as shown below to make Snort work.

```
output alert_syslog: LOG_AUTH LOG_ALERT
```
The option "alert_syslog" is used to log all Snort alerts to syslog on your server. This is what we have to use to get readable Snort reports.

```
include classification.config
```
The option "include classification.config" is used to inform Snort to include the "classification.config" file to its configuration. The Snort "classification.config" file is used to classify and prioritize alerts. We use it to specify what priority each classification should have. The default setting is suitable for all of us.

The rule set section

This section of the Snort configuration file is used to enable rules files that we hope to use with Snort. Most of the predefined rules are already enabled in the configuration file. Rules are command lines or signatures used to generate alerts based on suspicious activity. You can keep the default setting and Snort will work on your server. You can also get latest rules files from the Snort web site and update your rules if necessary. In general, you only need to comment or uncomment rules that you expect to use with Snort in this section.

As we said earlier, Snort uses rule sets to generate and get information about the way it should detect and interpret attacks on your network. Each common service has its own rule set available to use with Snort. In the configuration file, we use and enable all default Snort rules files except some that may provide false alarms. It is up to you to decide which additional rules you want to include with Snort. You can also write your own rule files to use since the software allows us to do it, but this is another story. Please see the Snort website for more information about how to create and use your own rules with Snort.

/etc/init.d/snort: The Snort Initialization File

The /etc/init.d/snort script file is responsible to automatically starting and stopping the Snort server on your Linux system. Please note that the following script is suitable for Linux operating systems that use SystemV. If you Linux system use some other methods like BSD, you'll have to adjust the script bellow to make it work for you.

Step 1
Create the **snort** script file (touch /etc/init.d/snort) and add the following lines inside it:

```
#!/bin/bash

# This shell script takes care of starting and stopping the snort IDS daemon.
#
# chkconfig: 2345 40 60
# description:  Snort is a lightweight network intrusion detection tool that \
#               currently detects more than 1100 host and network \
#               vulnerabilities, portscans, backdoors, and more.

# Source function library.
. /etc/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

# Specify your network interface here
INTERFACE=eth0

RETVAL=0
prog="Snort"

start() {
        echo -n $"Starting $prog: "
        daemon /usr/bin/snort -A fast -u snort -g snort -b -s -z -d -D \
                -i $INTERFACE -c /etc/snort/snort.conf
        RETVAL=$?
        echo
        [ $RETVAL = 0 ] && touch /var/lock/subsys/snort
        return $RETVAL
}

stop() {
        echo -n $"Shutting down $prog: "
        killproc snort
        RETVAL=$?
        echo
        [ $RETVAL = 0 ] && rm -f /var/lock/subsys/snort
        return $RETVAL
}

# See how we were called.
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
```

```
  status)
        status snort
        ;;
  restart)
        stop
        start
        ;;
  condrestart)
        [ -f /var/lock/subsys/snort ] && restart
        ;;
  *)
        echo $"Usage: $prog {start|stop|status|restart|condrestart}"
        exit 1
esac
exit $RETVAL
```

Step 2
Once the `/etc/init.d/snort` script file has been created, it is important to make it executable, change its default permissions, create the necessary links and start it. Making this file executable will allow the system to run it, changing its default permission to allow only the root user to change this file for security reasons, and creation of the symbolic links will let the process control initialization of Linux start the program automatically for you at each system boot.

- To make this script executable and to change its default permissions, use the command:
```
[root@deep /]# chmod 700 /etc/init.d/snort
[root@deep /]# chown 0.0 /etc/init.d/snort
```

- To create the symbolic `rc.d` links for `Snort`, use the following command:
```
[root@deep /]# chkconfig --add snort
[root@deep /]# chkconfig --level 2345 snort on
```

- To start `Snort` software manually, use the following command:
```
[root@deep /]# /etc/init.d/snort start
Starting Snort:          [OK]
```

Running `Snort` in a chroot jail

This section applies only if you want to run `Snort` in chroot jail environment. To do it, we need to create the required skeleton environment and copy necessary files into this chroot jail. Below are the steps to follow if you want to run `Snort` with chroot jail support.

The main benefit of a chroot jail is that the jail will limit the portion of the file system the daemon can see to the root directory of the jail. Additionally, since the jail only needs to support `Snort`, the programs available in the jail can be extremely limited. Most importantly, there is no need for setuid-root programs, which can be used to gain root access and break out of the jail.

Necessary steps to run Snort in a chroot jail:

What you're essentially doing is creating a skeleton root file system with just enough components necessary (files, directories) to allow UNIX to do a chroot when Snort starts. The procedures to run Snort in chroot jail are really easy to accomplish as follows.

Step 1
First, we have to create all the necessary chrooted environment subdirectories where we will move Snort files and directories.

- Use the following command to create all the necessary chroot subdirectories.

```
[root@deep /]# mkdir -p /chroot/snort/etc/snort
[root@deep /]# mkdir -p /chroot/snort/var/log/snort
[root@deep /]# chown -R snort.snort /chroot/snort/var/log/snort
```

Step 2
Now, it is time to move the required Snort files to the related subdirectories in the chroot area for Snort to work. We can copy these files to the chroot jail but it's better to move them to avoid unnecessary duplication of Snort files on the server.

- Use the following commands to move the require files into the chroot area.

```
[root@deep /]# mv /etc/snort/* /chroot/snort/etc/snort/
[root@deep /]# chmod 0644 /chroot/snort/etc/snort/*
```

Step 3
Once the Snort files have been moved to the chroot location, we can remove the old Snort directories from the system since they are no longer required.

- This can be done with the following commands.

```
[root@deep /]# rm -rf /etc/snort/
[root@deep /]# rm -rf /var/log/snort/
```

Step 4
Next, we have to recreate a new snort initialization script file which starts Snort in the chroot environment. Please note that the following script is suitable for Linux operating systems that use SystemV. If you Linux system use some other method like BSD, you'll have to adjust the script below to make it work for you. The only difference with the previous Snort initialization script file is that we use the "-t" option of Snort to specify the chroot location.

Edit the **snort** script file (vi /etc/init.d/snort) and add the following lines inside it:

```
#!/bin/bash

# This shell script takes care of starting and stopping the snort IDS daemon.
#
# chkconfig: 2345 40 60
# description:  Snort is a lightweight network intrusion detection tool that \
#               currently detects more than 1100 host and network \
#               vulnerabilities, portscans, backdoors, and more.

# Source function library.
. /etc/init.d/functions

# Source networking configuration.
```

```
. /etc/sysconfig/network

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

# Specify your network interface here
INTERFACE=eth0

RETVAL=0
prog="Snort"

start() {
        echo -n $"Starting $prog: "
        daemon /usr/bin/snort -A fast -u snort -g snort -b -s -z -d -D \
                -i $INTERFACE -c /etc/snort/snort.conf -t /chroot/snort/
        RETVAL=$?
        echo
        [ $RETVAL = 0 ] && touch /var/lock/subsys/snort
        return $RETVAL
}

stop() {
        echo -n $"Shutting down $prog: "
        killproc snort
        RETVAL=$?
        echo
        [ $RETVAL = 0 ] && rm -f /var/lock/subsys/snort
        return $RETVAL
}

# See how we were called.
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  status)
        status snort
        ;;
  restart)
        stop
        start
        ;;
  condrestart)
        [ -f /var/lock/subsys/snort ] && restart
        ;;
  *)
        echo $"Usage: $prog {start|stop|status|restart|condrestart}"
        exit 1
esac
exit $RETVAL
```

Step 5
Finally, we must test the new chrooted jail configuration of our Snort program.

- Start the new chrooted jail Snort with the following command:
```
[root@deep /]# /etc/init.d/snort start
Starting Snort:                          [OK]
```

- If you don't get any errors, do a **ps ax | grep snort** and see if we're running:
```
[root@deep /]# ps ax | grep snort
16295 ?   R   0:38 /usr/bin/snort -A fast -u snort -g snort -b -s -z -d
```

If so, lets check to make sure it's chrooted by picking its process number and doing **ls -la /proc/that_process_number/root/**.

```
[root@deep /]# ls -la /proc/16295/root/
```

If you see something like:
```
total 4
drwxr-xr-x    4 root     root         4096 May  7 19:10 ./
drwxr-xr-x    5 root     root         4096 May  7 19:12 ../
drwxr-xr-x    3 root     root         4096 May  7 19:12 etc/
drwxr-xr-x    3 root     root         4096 May  7 19:12 var/
```

Congratulations! Your Snort in a chroot jail is working.

Further documentation
For more details, there is one manual page about Snort that you should read:

```
$ man snort (8)          - Open source network intrusion detection system.
```

CHAPTER 23

Tripwire

IN THIS CHAPTER

1. Compiling - Optimizing & Installing `Tripwire`
2. Configuring `Tripwire`
3. Running `Tripwire` for the first time
4. Securing `Tripwire`
5. `Tripwire` Administrative Tools

Linux Tripwire

Abstract
With the advent of increasingly sophisticated and subtle account break-ins on Unix systems, the need for tools to aid in the detection of unauthorized modification of files becomes clear.

Tripwire is a tool that aids system administrators and users in monitoring a designated set of files for any changes. Used with system files on a regular (e.g., daily) basis, Tripwire can notify system administrators of corrupted or tampered files, so damage control measures can be taken in a timely manner.

Tripwire data and network integrity software was originally developed in 1992 at Purdue University by world-renowned computer security expert, Dr. Eugene Spafford, and by master's degree student, Gene Kim. It was quickly embraced by computer security experts and actively used by thousands of corporate, government, and educational organizations worldwide.

Tripwire is a file and directory integrity checker, a utility that compares a designated set of files and directories against information stored in a previously generated database. Any differences are flagged and logged, including added or deleted entries.

When run against system files on a regular basis, any changes in critical system files will be spotted -- and appropriate damage control measures can be taken immediately. With Tripwire, system administrators can conclude with a high degree of certainty that a given set of files remain free of unauthorized modifications if Tripwire reports no changes.

Tripwire is a very valuable security tool for Linux systems, if it is installed to a clean system. Tripwire should be installed right after the OS installation, and before you have connected your system to a network (i.e., before any possibility exists that someone could alter files on your system).

These installation instructions assume
Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest Tripwire version number is 2.3.1-2

The procedures given in this chapter are likely to work on all Linux platforms, but we have only tested it on OpenNA Linux and Red Hat Linux.

Packages
The following is based on information listed by Tripwire as of 2001/03/03. Please check http://sourceforge.net/projects/tripwire/ regularly for the latest status. We chose to install from source because it provides the facility to fine tune the installation.

Source code is available from:
Tripwire Homepage: http://sourceforge.net/projects/tripwire/
You must be sure to download: tripwire-2.3.1-2.tar.gz

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all the files installed onto the system if you want to update the package in the future. To solve this problem, it's a good idea to make a list of files on the system before you install Tripwire, and then one afterwards, and then compare them using the **diff** utility to find out what files were placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep root]# find /* > Tripwire1
  ```

- And the following one after you install the software:
  ```
  [root@deep root]# find /* > Tripwire2
  ```

- Then use the following command to get a list of what changed:
  ```
  [root@deep root]# diff Tripwire1 Tripwire2 > Tripwire-Installed
  ```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. In our example above, we use the /root directory of the system to store all the generated file lists.

Compiling - Optimizing & Installing Tripwire

Below are the steps that you must make to configure, compile and optimize the Tripwire software before installing it on your system. First off, we install the program as user 'root' so as to avoid any authorization problems.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- This can be done with the following commands:
  ```
  [root@deep /]# cp tripwire-version.tar.gz /var/tmp/
  [root@deep /]# cd /var/tmp/
  [root@deep tmp]# tar xzpf tripwire-version.tar.gz
  ```

Step 2
There are some source files to modify before going into the configuration and compilation of the program; the changes allow us to fix many bugs with Tripwire. Therefore, move into the newly created Tripwire source directory and perform the following steps to configure and optimize the software for your system.

- To move into the newly created Tripwire directory use the following command:
  ```
  [root@deep tmp]# cd tripwire-2.3.1-2/
  ```

Step 3
The first source file to modify is called "mailmessage.cpp".

- Edit the **mailmessage.cpp** file (vi +244 src/tripwire/mailmessage.cpp) and change:

```
const TCHAR* szFormat = _T("%a %d %b %Y %H:%M:%S %z");
```

To read:

```
const TCHAR* szFormat = _T("%a, %d %b %Y %H:%M:%S %z");
```

Step 4
The second file is called "platform.h" and we have to edit it and add a new line as follows.

- Edit the **platform.h** file (vi +294 src/core/platform.h) and change the line:

```
#define USES_FHS                                IS_LINUX
```

To read:

```
#define USE_FHS                                 IS_LINUX
```

Step 5
The last file to modify is very important for Linux systems with GCC version 3; which should be the default compiler for most Linux system now. The modifications are important to allow Tripwire to compile with GCC v3. There is one problem, the modifications are too big to be listed in a book and we have to retrieve it from the Internet as a patch file and patch our sources code.

The patch is available from the OpenNA website at the following URL:
http://www.openna.com/products/books/securing-optimizing-linux/3rdedition/index.htm
Please, download the patch and patch your Tripwire source codes as follow:

- To patch your Tripwire source codes, use the command:

```
[root@deep /]# cp tripwire-gcc3.patch /var/tmp/
[root@deep /]# cd /var/tmp/tripwire-2.3.1-2/
[root@deep tripwire-2.3.1-2]# patch -p1 < ../tripwire-gcc3.patch
```

Step 6
Now, we must make a list of all files on the system before installing the software, and one afterwards, then compare them using the **diff** utility to find out what files are placed where and finally we install the Tripwire software:

```
[root@deep tripwire-2.3.1-2]# cd src/
[root@deep src]# rm -rf STLport*
[root@deep src]# touch STLport_r STLport_d
[root@deep src]# export CXXFLAGS="-O2 -march=i686 -funroll-loops"
[root@deep src]# make release
[root@deep src]# cd
[root@deep root]# find /* > Tripwire1
[root@deep root]# cd /var/tmp/tripwire-2.3.1-2/bin/i686-pc-linux_r/
[root@deep i686-pc-linux_r]# install -m0500 siggen /usr/sbin/
[root@deep i686-pc-linux_r]# install -m0500 tripwire /usr/sbin/
[root@deep i686-pc-linux_r]# install -m0500 twadmin /usr/sbin/
[root@deep i686-pc-linux_r]# install -m0500 twprint /usr/sbin/
[root@deep i686-pc-linux_r]# cd ../../man/
[root@deep man]# install -m0440 man4/*.4 /usr/share/man/man4/
[root@deep man]# install -m0440 man5/*.5 /usr/share/man/man5/
[root@deep man]# install -m0440 man8/*.8 /usr/share/man/man8/
[root@deep man]# mkdir -m0700 /etc/tripwire
[root@deep man]# mkdir -p /var/lib/tripwire/report
[root@deep man]# chmod -R 0700 /var/lib/tripwire/
[root@deep man]# cd
[root@deep root]# find /* > Tripwire2
[root@deep root]# diff Tripwire1 Tripwire2 > Tripwire-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

Step 7
Once the configuration, optimization, compilation, and installation of the Tripwire software have been accomplished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete Tripwire and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf tripwire-version/
[root@deep tmp]# rm -f tripwire-version.tar.gz
```

The rm command as used above will remove all the source files we have used to compile and install Tripwire. It will also remove the Tripwire compressed archive from the /var/tmp directory.

Configuring Tripwire

After Tripwire has been built and installed successfully in your system, your next step is to configure and customize its configuration files and policy file to fit your needs.

- ✓ /etc/tripwire/twcfg.txt: (The Tripwire Configuration File)
- ✓ /etc/tripwire/twpol.txt: (The Tripwire Policy File)
- ✓ /etc/tripwire/twinstall.sh: (The Tripwire Cryptographic File)
- ✓ /etc/cron.weekly/tripwire.cron: (The Tripwire Cron File)

/etc/tripwire/twcfg.txt: The Tripwire Configuration File

The twcfg.txt file is a small Tripwire configuration file used by the program the first time you run it to get information about locations of different files and the way Tripwire runs and reports on the integrity of the system. It stores system-specific information, such as the location of Tripwire data files. In general, we use this file ONE time and REMOVE it from the server once Tripwire is configured.

Step 1
By default, the twcfg.txt file do not exist after installation, we have to create it as follow.

- Create the **twcfg.txt** file (touch /etc/tripwire/twcfg.txt) and add the following lines inside the file. Below is what we recommend you.

```
ROOT                    =/usr/sbin
POLFILE                 =/etc/tripwire/tw.pol
DBFILE                  =/var/lib/tripwire/$(HOSTNAME).twd
REPORTFILE              =/var/lib/tripwire/report/$(HOSTNAME)-$(DATE).twr
SITEKEYFILE             =/etc/tripwire/site.key
LOCALKEYFILE            =/etc/tripwire/$(HOSTNAME)-local.key
EDITOR                  =/bin/vi
LATEPROMPTING           =true
LOOSEDIRECTORYCHECKING =true
MAILNOVIOLATIONS        =false
EMAILREPORTLEVEL        =3
REPORTLEVEL             =3
MAILMETHOD              =SENDMAIL
SYSLOGREPORTING         =true
MAILPROGRAM             =/usr/sbin/sendmail -oi -t
```

Step2
Now, set the permissions of the twcfg.txt file to be (0640/-rw-r-----) and owned by the super-user 'root' for security reasons.

- To change the permission mode and ownership of the twcfg.txt file, use:

```
[root@deep /]# chmod 640 /etc/tripwire/twcfg.txt
[root@deep /]# chown 0.0 /etc/tripwire/twcfg.txt
```

/etc/tripwire/twpol.txt: The Tripwire Policy File

The twpol.txt file is the Tripwire policy file where you decide and set which system files and directories that you want monitored. It consists of a series of rules specifying the system objects the Tripwire should monitor, and the data for each object that should be collected and stored in the database files. Note that extensive testing and experience are necessary when editing this file before you get working file reports. The following is a working example from where you can start you own customization. We must create, edit or change it to fit our requirements and operating system.

Step 1
By default, the twpol.txt file does not exist after installation; we have to create it as follow. The text in bold are the parts of the configuration file that must be customized and adjusted to fit your own system.

- Create the **twpol.txt** file (touch /etc/tripwire/twpol.txt) and add in this file all the files and directories that you want monitored. The format of the configuration file is described in its header and in the manual page twpolicy (4). Below is what we recommend you enter:

```
# This is the example Tripwire Policy file. It is intended as a place to
# start creating your own custom Tripwire Policy file. Referring to it as
# well as the Tripwire Policy Guide should give you enough information to
# make a good custom Tripwire Policy file that better covers your
# configuration and security needs.
#
# Because it is impossible to be one policy file for all machines, your
# Linux configuration will most likey differ from the one our policy file
# was tuned to, and will therefore require some editing of the default
# Tripwire Policy file.

# Global Variable Definitions
# These are defined at install time by the installation script. You may
# Manually edit these if you are using this file directly and not from the
# installation script itself.

@@section GLOBAL
TWROOT=/usr/sbin;
TWBIN=/usr/sbin;
TWPOL="/etc/tripwire";
TWDB="/var/lib/tripwire";
TWSKEY="/etc/tripwire";
TWLKEY="/etc/tripwire";
TWREPORT="/var/lib/tripwire/report";

# NOTE: Change the following parameter to reflect your own host name.
# For example, if your host is called 'www', then change 'localhost' to 'www'.
HOSTNAME=localhost;

@@section FS
SEC_CRIT      = $(IgnoreNone)-SHa ;
SEC_SUID      = $(IgnoreNone)-SHa ;
SEC_BIN       = $(ReadOnly) ;
SEC_CONFIG    = $(Dynamic) ;
SEC_LOG       = $(Growing) ;
SEC_INVARIANT = +tpug ;
SIG_LOW       = 33 ;
SIG_MED       = 66 ;
SIG_HI        = 100 ;
```

```
# SEC_CRIT are critical files that cannot change.
# SEC_SUID are binaries with the SUID or SGID flags set.
# SEC_BIN are binaries that should not change.
# SEC_CONFIG are config files that are changed infrequently but accessed often.
# SEC_LOG are files that grow, but that should never change ownership.
# SEC_INVARIANT are directories that should never change permission or owners.
# SIG_LOW are non-critical files that are of minimal security impact.
# SIG_MED are non-critical files that are of significant security impact.
# SIG_HI are critical files that are significant points of vulnerability.


(
  rulename = "Tripwire binaries",
  severity = $(SIG_HI)
)
{
  $(TWBIN)/siggen                       -> $(SEC_BIN) ;
  $(TWBIN)/tripwire                     -> $(SEC_BIN) ;
  $(TWBIN)/twadmin                      -> $(SEC_BIN) ;
  $(TWBIN)/twprint                      -> $(SEC_BIN) ;
}


(
  rulename = "Tripwire data files",
  severity = $(SIG_HI)
)
{
  $(TWDB)                               -> $(SEC_CONFIG) -i ;
  $(TWPOL)/tw.pol                       -> $(SEC_BIN) -i ;
  $(TWPOL)/tw.cfg                       -> $(SEC_BIN) -i ;
  $(TWLKEY)/$(HOSTNAME)-local.key       -> $(SEC_BIN) ;
  $(TWSKEY)/site.key                    -> $(SEC_BIN) ;
  $(TWREPORT)                           -> $(SEC_CONFIG) (recurse=0) ;
}


(
  rulename = "Invariant directories",
  severity = $(SIG_MED)
)
{
  /                                     -> $(SEC_INVARIANT) (recurse = 0) ;
  /home                                 -> $(SEC_INVARIANT) (recurse = 0) ;
}


(
  rulename = "/root directory",
  severity = $(SIG_HI)
)
{
  /root                                 -> $(SEC_CRIT) (recurse = -1) ;
  /root/.bashrc                         -> $(SEC_CONFIG) (recurse = 0) ;
  /root/.bash_profile                   -> $(SEC_CONFIG) (recurse = 0) ;
  /root/.bash_logout                    -> $(SEC_CONFIG) (recurse = 0) ;
  /root/.bash_history                   -> $(SEC_CONFIG) (recurse = 0) ;
}


(
  rulename = "/boot directory",
  severity = $(SIG_HI)
```

```
)
{
  /boot                                -> $(SEC_CRIT) (recurse = -1) ;
  !/boot/System.map ;
}


(
  rulename = "/etc directory",
  severity = $(SIG_HI)
)
{
  /etc                                 -> $(SEC_CRIT) (recurse = -1) ;
}


(
  rulename = "/dev & /proc directories",
  severity = $(SIG_HI),
)
{
  /dev                                 -> $(Device) (recurse = -1) ;
  /proc/bus                            -> $(Device) (recurse = 0) ;
  /proc/cmdline                        -> $(Device) (recurse = 0) ;
  /proc/cpuinfo                        -> $(Device) (recurse = 0) ;
  /proc/devices                        -> $(Device) (recurse = 0) ;
  /proc/dma                            -> $(Device) (recurse = 0) ;
  /proc/driver                         -> $(Device) (recurse = 0) ;
  /proc/execdomains                    -> $(Device) (recurse = 0) ;
  /proc/filesystems                    -> $(Device) (recurse = 0) ;
  /proc/fs                             -> $(Device) (recurse = 0) ;
  /proc/ide                            -> $(Device) (recurse = 0) ;
  /proc/interrupts                     -> $(Device) (recurse = 0) ;
  /proc/iomem                          -> $(Device) (recurse = 0) ;
  /proc/ioports                        -> $(Device) (recurse = 0) ;
  /proc/irq                            -> $(Device) (recurse = 0) ;
  /proc/kcore                          -> $(Device) (recurse = 0) ;
  /proc/kmsg                           -> $(Device) (recurse = 0) ;
  /proc/ksyms                          -> $(Device) (recurse = 0) ;
  /proc/loadavg                        -> $(Device) (recurse = 0) ;
  /proc/locks                          -> $(Device) (recurse = 0) ;
  /proc/meminfo                        -> $(Device) (recurse = 0) ;
  /proc/misc                           -> $(Device) (recurse = 0) ;
  /proc/mounts                         -> $(Device) (recurse = 0) ;
  /proc/partitions                     -> $(Device) (recurse = 0) ;
  /proc/pci                            -> $(Device) (recurse = 0) ;
  /proc/self                           -> $(Device) (recurse = 0) ;
  /proc/slabinfo                       -> $(Device) (recurse = 0) ;
  /proc/stat                           -> $(Device) (recurse = 0) ;
  /proc/sys                            -> $(Device) (recurse = 0) ;
  /proc/sysvipc                        -> $(Device) (recurse = 0) ;
  /proc/tty                            -> $(Device) (recurse = 0) ;
  /proc/uptime                         -> $(Device) (recurse = 0) ;
  /proc/version                        -> $(Device) (recurse = 0) ;
  !/dev/pts ;
  !/dev/shm ;
}


(
  rulename = "/bin & /sbin directories",
  severity = $(SIG_HI)
)
```

```
{
  /bin                                -> $(SEC_CRIT) (recurse = -1) ;
  /sbin                               -> $(SEC_CRIT) (recurse = -1) ;
}


(
  rulename = "/lib directory",
  severity = $(SIG_HI)
)
{
  /lib                                -> $(SEC_CRIT) (recurse = -1) ;
}


(
  rulename = "/tmp directories",
  severity = $(SIG_LOW)
)
{
  /usr/tmp                            -> $(SEC_INVARIANT) (recurse = 0) ;
  /var/tmp                            -> $(SEC_INVARIANT) (recurse = 0) ;
  /tmp                                -> $(SEC_INVARIANT) (recurse = 0) ;
}


(
  rulename = "/urs directories",
  severity = $(SIG_HI)
)
{
  /usr                                -> $(SEC_CRIT) (recurse = -1) ;
}


(
  rulename = "/var directories",
  severity = $(SIG_HI)
)
{
  /var                                -> $(SEC_CONFIG) (recurse = -1) ;
  /var/lib                            -> $(SEC_CONFIG) (recurse = -1) ;
  /var/spool                          -> $(SEC_CONFIG) (recurse = -1) ;
  !/var/spool/mail ;
  !/var/spool/mqueue ;
}


(
  rulename = "SUID SGID binaries",
  severity = $(SIG_HI)
)
{
  /usr/bin/man                        -> $(SEC_SUID) (recurse = 0) ;
  /usr/bin/slocate                    -> $(SEC_SUID) (recurse = 0) ;
  /usr/bin/passwd                     -> $(SEC_SUID) (recurse = 0) ;
  /usr/bin/crontab                    -> $(SEC_SUID) (recurse = 0) ;
  /usr/bin/sudo                       -> $(SEC_SUID) (recurse = 0) ;
  /usr/sbin/utempter                  -> $(SEC_SUID) (recurse = 0) ;
  /usr/sbin/exim                      -> $(SEC_SUID) (recurse = 0) ;
  /bin/su                             -> $(SEC_SUID) (recurse = 0) ;
}
```

```
(
  rulename = "/chroot directory",
  severity = $(SIG_HI)
)
{
  /chroot                                -> $(SEC_CRIT) (recurse = -1) ;
}
```

Step2
Now, set the permissions of the `twpol.txt` file to be (`0640/-rw-r-----`) and owned by the super-user '`root`' for security reasons.

- To change the permission mode and ownership of the `twpol.txt` file, use:
 `[root@deep /]#` **`chmod 640 /etc/tripwire/twpol.txt`**
 `[root@deep /]#` **`chown 0.0 /etc/tripwire/twpol.txt`**

NOTE: Please, add to the above policy file all files, binaries, directories that you want the software to monitor for you. Remove any files, binaries, directories that you don't want the software to monitor for you and don't send emails to the mailing list if you receive error messages about the fact that some files, binaries, directories don't exist on your system. Instead, review your policy file and make the changes related to the error that you received, because in most cases, this is why you have this kind of errors messages. Finally, reads the `twpolicy` manual page for more information on the parameters of this policy file.

`/etc/tripwire/twinstall.sh`: The `Tripwire` Cryptographic File

The `twinstall.sh` file is a script file used by `Tripwire` to configure, install and generate cryptographic keys used by `Tripwire` during operation and verification.

Step 1
This script file asks you the passpharse that you want to run with `Tripwire` as well as other operations related to the `Tripwire` database generation and policies. By default, the `twinstall.sh` file does not exist after the installation; we have to create it as follows.

- Create the **`twinstall.sh`** file (`touch /etc/tripwire/twinstall.sh`) and add the following lines inside the file.

```
#!/bin/sh

HOST_NAME='localhost'
if uname -n > /dev/null 2> /dev/null ; then
        HOST_NAME=`uname -n`
fi

# Site Passphrase variable
TW_SITE_PASS=""

# Complete path to site key
SITE_KEY="/etc/tripwire/site.key"

# Local Passphrase variable
TW_LOCAL_PASS=""

# Complete path to local key
```

```
LOCAL_KEY="/etc/tripwire/${HOST_NAME}-local.key"

# If clobber==true, overwrite files; if false, do not overwrite files.
CLOBBER="false"

# If prompt==true, ask for confirmation before continuing with install.
PROMPT="true"

# Name of twadmin executeable
TWADMIN="twadmin"

# Path to twadmin executeable
TWADMPATH=@sbindir@

# Path to configuration directory
CONF_PATH="/etc/tripwire"

# Name of clear text policy file
TXT_POL=$CONF_PATH/twpol.txt

# Name of clear text configuration file
TXT_CFG=$CONF_PATH/twcfg.txt

# Name of encrypted configuration file
CONFIG_FILE=$CONF_PATH/tw.cfg

# Path of the final Tripwire policy file (signed)
SIGNED_POL=`grep POLFILE $TXT_CFG | sed -e 's/^.*=\(.*\)/\1/'`

if [ -z "$TW_SITE_PASS" ] || [ -z "$TW_LOCAL_PASS" ]; then
cat << END_OF_TEXT

-----------------------------------------------
The Tripwire site and local passphrases are used to
sign a variety of files, such as the configuration,
policy, and database files.

Passphrases should be at least 8 characters in length
and contain both letters and numbers.

See the Tripwire manual for more information.
END_OF_TEXT
fi

echo
echo "-----------------------------------------------"
echo "Creating key files..."

if [ "$CLOBBER" = "true" ] && [ "$PROMPT" = "false" ] && [ -f "$SITE_KEY" ] ;
then
        rm -f "$SITE_KEY"
fi

if [ -f "$SITE_KEY" ] && [ "$CLOBBER" = "false" ] ; then
        echo "The site key file \"$SITE_KEY\""
        echo 'exists and will not be overwritten.'
else
        cmdargs="--generate-keys --site-keyfile \"$SITE_KEY\""
        if [ -n "$TW_SITE_PASS" ] ; then
                cmdargs="$cmdargs --site-passphrase \"$TW_SITE_PASS\""
        fi
        eval "\"$TWADMPATH/$TWADMIN\" $cmdargs"
        if [ $? -ne 0 ] ; then
```

```
                echo "Error: site key generation failed"
                exit 1
        else chmod 640 "$SITE_KEY"
        fi
fi

if [ "$CLOBBER" = "true" ] && [ "$PROMPT" = "false" ] && [ -f "$LOCAL_KEY" ] ;
then
        rm -f "$LOCAL_KEY"
fi

if [ -f "$LOCAL_KEY" ] && [ "$CLOBBER" = "false" ] ; then
        echo "The site key file \"$LOCAL_KEY\""
        echo 'exists and will not be overwritten.'
else
        cmdargs="--generate-keys --local-keyfile \"$LOCAL_KEY\""
        if [ -n "$TW_LOCAL_PASS" ] ; then
                cmdargs="$cmdargs --local-passphrase \"$TW_LOCAL_PASS\""
        fi
        eval "\"$TWADMPATH/$TWADMIN\" $cmdargs"
        if [ $? -ne 0 ] ; then
                echo "Error: local key generation failed"
                exit 1
        else chmod 640 "$LOCAL_KEY"
        fi
fi

echo
echo "-----------------------------------------------"
echo "Signing configuration file..."

if [ "$CLOBBER" = "false" ] && [ -s "$CONFIG_FILE" ] ; then
        backup="${CONFIG_FILE}.$$.bak"
        echo "Backing up $CONFIG_FILE"
        echo "        to $backup"
        `mv "$CONFIG_FILE" "$backup"`
        if [ $? -ne 0 ] ; then
                echo "Error: backup of configuration file failed."
                exit 1
        fi
fi

cmdargs="--create-cfgfile"
cmdargs="$cmdargs --cfgfile \"$CONFIG_FILE\""
cmdargs="$cmdargs --site-keyfile \"$SITE_KEY\""
if [ -n "$TW_SITE_PASS" ] ; then
        cmdargs="$cmdargs --site-passphrase \"$TW_SITE_PASS\""
fi

eval "\"$TWADMPATH/$TWADMIN\" $cmdargs \"$TXT_CFG\""
if [ $? -ne 0 ] ; then
        echo "Error: signing of configuration file failed."
        exit 1
fi

# Set the rights properly
chmod 640 "$CONFIG_FILE"

cat << END_OF_TEXT

A clear-text version of the Tripwire configuration file
$TXT_CFG
has been preserved for your inspection.  It is recommended
```

```
that you delete this file manually after you have examined it.

END_OF_TEXT

echo
echo "-----------------------------------------------"
echo "Signing policy file..."

if [ "$CLOBBER" = "false" ] && [ -s "$POLICY_FILE" ] ; then
        backup="${POLICY_FILE}.$$.bak"
        echo "Backing up $POLICY_FILE"
        echo "        to $backup"
        mv "$POLICY_FILE" "$backup"
        if [ $? -ne 0 ] ; then
                echo "Error: backup of policy file failed."
                exit 1
        fi
fi

cmdargs="--create-polfile"
cmdargs="$cmdargs --cfgfile \"$CONFIG_FILE\""
cmdargs="$cmdargs --site-keyfile \"$SITE_KEY\""
if [ -n "$TW_SITE_PASS" ] ; then
        cmdargs="$cmdargs --site-passphrase \"$TW_SITE_PASS\""
fi

eval "\"$TWADMPATH/$TWADMIN\" $cmdargs \"$TXT_POL\""
if [ $? -ne 0 ] ; then
        echo "Error: signing of policy file failed."
        exit 1
fi

# Set the proper rights on the newly signed policy file.
chmod 0640 "$SIGNED_POL"

cat << END_OF_TEXT

A clear-text version of the Tripwire policy file
$TXT_POL
has been preserved for your inspection.  This implements
a minimal policy, intended only to test essential
Tripwire functionality.  You should edit the policy file
to describe your system, and then use twadmin to generate
a new signed copy of the Tripwire policy.

END_OF_TEXT
```

Step 2
Now, set the permissions of the `twinstall.sh` file to be (`0500/---x------`) and owned by the super-user '`root`' for security reasons.

- This procedure can be accomplished with the following command:
  ```
  [root@deep /]# chmod 500 /etc/tripwire/twinstall.sh
  [root@deep /]# chown 0.0 /etc/tripwire/twinstall.sh
  ```

NOTE: The above script file can also be retrieved from the following URL:
http://www.openna.com/products/books/securing-optimizing-linux/3rdedition/index.htm

/etc/cron.weekly/tripwire.cron: The Tripwire Cron File

The tripwire.cron file is a small script executed automatically by the crond program each week to scan your hard disk for possible changes to files or directories and mail the results to the system administrator.

Step 1
This script will automate the procedure of integrity checking for you. If you want to automate this task, follow the simple steps below.

- Create the **tripwire.cron** script file (touch /etc/cron.weekly/tripwire.cron) and add the following lines:

```
#!/bin/sh
HOST_NAME=`uname -n`
if [ ! -e /var/lib/tripwire/${HOST_NAME}.twd ] ; then
  echo "****    Error: Tripwire database for ${HOST_NAME} not found.    ****"
  echo "**** Run "/etc/tripwire/twinstall.sh" and/or "tripwire --init". ****"
else
        test -f /etc/tripwire/tw.cfg &&  /usr/sbin/tripwire --check
fi
```

Step 2
Now, set the permissions of the tripwire.cron file to be (0500/---x------) and owned by the super-user 'root' for security reasons.

- This procedure can be accomplished with the following command:

```
[root@deep /]# chmod 500 /etc/cron.weekly/tripwire.cron
[root@deep /]# chown 0.0 /etc/cron.weekly/tripwire.cron
```

Running Tripwire for the first time

Once all the files are installed and configured on your server, it's time to run Tripwire to generate the cryptography key, passphrases and database files. These procedures should be made the first time you install the software and ONLY the first time.

Step 1
Here we begin by running the twinstall.sh script file which will generate the cryptography keys and will ask us to enter our passphrase (password) which is required each time we want to update and accept Tripwire integrity reports.

- To run the twinstall.sh file use the following command:

```
[root@deep /]# /etc/tripwire/twinstall.sh
-----------------------------------------------
The Tripwire site and local passphrases are used to
sign a variety of files, such as the configuration,
policy, and database files.

Passphrases should be at least 8 characters in length
and contain both letters and numbers.

See the Tripwire manual for more information.

-----------------------------------------------
```

```
Creating key files...

(When selecting a passphrase, keep in mind that good passphrases
typically have upper and lower case letters, digits and punctuation
marks, and are at least 8 characters in length.)

Enter the site keyfile passphrase: Your site keyfile passphrase
Verify the site keyfile passphrase: Your site keyfile passphrse again
Generating key (this may take several minutes)...Key generation complete.

(When selecting a passphrase, keep in mind that good passphrases
typically have upper and lower case letters, digits and punctuation
marks, and are at least 8 characters in length.)

Enter the local keyfile passphrase: Your local keyfile passphrase
Verify the local keyfile passphrase: Your local keyfile passphrase again
Generating key (this may take several minutes)...Key generation complete.

---------------------------------------------
Signing configuration file...
Please enter your site passphrase: Your site passphrase
Wrote configuration file: /etc/tripwire/tw.cfg

A clear-text version of the Tripwire configuration file
/etc/tripwire/twcfg.txt
has been preserved for your inspection.  It is recommended
that you delete this file manually after you have examined it.

---------------------------------------------
Signing policy file...
Please enter your site passphrase: Your site passphrase
Wrote policy file: /etc/tripwire/tw.pol

A clear-text version of the Tripwire policy file
/etc/tripwire/twpol.txt
has been preserved for your inspection. This implements
a minimal policy, intended only to test essential
Tripwire functionality. You should edit the policy file
to describe your system, and then use twadmin to generate
a new signed copy of the Tripwire policy
```

Step 2

Once our passphrase keyfiles have been generated, it's time to run `Tripwire` in its' initialization mode. The initialization mode will create the initial `Tripwire` database files based on what information has been provided inside the `twpol.txt` file. `Tripwire` must have a database to compare against, so we first create the file information database. This action will create a file called "`tw.db_[hostname]`" in the directory you specified to hold your databases (where `[hostname]` will be replaced with your machine hostname).

- To run the `Tripwire` in initialization mode, use the following command:

```
[root@deep /]# tripwire --init
Parsing policy file: /etc/tripwire/tw.pol
Generating the database...
*** Processing Unix File System ***
Please enter your local passphrase:
Wrote database file: /var/lib/tripwire/deep.twd
The database was successfully generated.
```

NOTE: Initialization of the database `Tripwire` uses should be done manually because the key used to sign the database should be different for each system.

Step 3
Finally, if you have not received any kind of error message, then you can safety remove the `twcfg.txt` and `twpol.txt` files from your system since they are no longer needed and it would be a security risk to keep these files on your server.

- To remove the files from your system, use the following commands:
```
[root@deep /]# rm -f /etc/tripwire/twcfg.txt
[root@deep /]# rm -f /etc/tripwire/twpol.txt
```

NOTE: You have to remove the files from your server ONLY if you are sure that the initialization of the databases has been completed without any errors. Otherwise you should keep these files and regenerate a new database once all the errors have been fixed inside the `twpol.txt` file, since in many cases errors come from `twpol.txt` file having some lines referring to files or directories that do not exist in your system.

Securing `Tripwire`

It is highly recommended that the database (`tw.db_[hostname]`) file of `Tripwire` be moved someplace (e.g. floppy) where it cannot be modified. This is important because data from `Tripwire` is only as trustworthy as its database.

It is also recommended that you make a hardcopy printout of the database contents right away. In the event that you become suspicious of the integrity of the database, you will be able to manually compare information against this hardcopy.

`Tripwire` Administrative Tools

The commands listed below are some of the most used of this software, but many more exist. Check the `Tripwire` manual pages for more details.

Running `Tripwire` in Interactive Checking Mode:

In “Interactive Checking Mode” feature, `Tripwire` verifies files or directories that have been added, deleted, or changed from the original database and ask the user whether the database entry should be updated. This mode is the most convenient way of keeping your database up-to-date, but it requires that the user be "at the console". If you want to use this mode, then follow the simple step below.

Once the file information database of `Tripwire` has been created, we can now run `Tripwire` in “Interactive Checking Mode”. This mode will prompt the user for whether or not each changed entry on the system should be updated to reflect the current state of the file.

- To run `Tripwire` in Interactive Checking Mode, use the following command:
```
[root@deep /]# tripwire --check --interactive
Parsing policy file: /etc/tripwire/tw.pol
*** Processing Unix File System ***
Performing integrity check...
```

NOTE: In interactive mode, `Tripwire` first reports all added, deleted, or changed files, and then allows the user to update the entry in the database.

Further documentation

For more details, there are several manual pages about `Tripwire` that you can read:

```
$ man siggen (8)          - Signature gathering routine for Tripwire.
$ man tripwire (8)        - A file integrity checker for UNIX systems.
$ man twadmin (8)         - Tripwire administrative and utility tool.
$ man twintro (8)         - Introduction to Tripwire software.
$ man twprint (8)         - Tripwire database and report printer.
$ man twconfig (4)        - Tripwire configuration file reference.
$ man twpolicy (4)        - Tripwire policy file reference.
$ man twfiles (5)         - Overview of files used by Tripwire and file backup process.
```

Some possible uses of `Tripwire` software

`Tripwire` can be used to:

1. Check the integrity of your files system.
2. Get a list of new installed or removed files on your system.



Part VI

Super-Server

CHAPTER 24

ucspi-tcp

IN THIS CHAPTER

1. Compiling - Optimizing & Installing `ucsip-tcp`
2. Using `ucsip-tcp`

Linux ucspi-tcp

Abstract

UCSPI stand for (**U**NIX **C**lient-**S**erver **P**rogram **I**nterface) and it's a command-line interface to client-server communications tools that provides several small programs like tcpserver or tcpclient, which are easy-to-use command-line tools for building TCP client-server applications.

Some may ask why we would need to run this kind of program on our server. Well, in the UNIX world, there is some software that cannot run as a daemon and need the help of other software like ucspi-tcp to work.

This is where ucspi-tcp is required. This small piece of software from D. J. Bernstein provides two important binary programs to achieve this. The first is called "tcpserver", which waits for incoming connections and, for each connection, runs a program of your choice, the second is called "tcpclient", which makes a TCP connection and runs a program of your choice.

Other tools exist in this ucspi-tcp package but the most frequently used are tcpserver and tcpclient. In general, we use these programs to replace software like inetd or Xinetd, which perform the same functions as tcpserver and tcpclient.

The main difference is that ucsip-tcp is really the most secure and faster software in this group. Personally, and each time you need to run third party software like IMAP, POP3, Qmail, vsFTPd, etc that depends on a super-server to work, I highly recommend you use ucspi-tcp instead of inet or Xinetd. That's said; let's go to the most interesting part now.

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest ucsip-tcp version number is 0.88

The procedures given in this chapter are likely to work on all Linux platforms, but we have only tested it on OpenNA Linux and Red Hat Linux.

Packages

The following is based on information listed by ucsip-tcp as of 2002/04/19. Please regularly check at http://cr.yp.to/ucspi-tcp/install.html for the latest status. We chose to install the required components from source because it provides the facility to fine tune the installation.

Source code is available from:
ucsip-tcp Homepage: http://cr.yp.to/ucspi-tcp.html
You must be sure to download: ucspi-tcp-0.88.tar.gz

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all the installed files in the system in the event of an update in the future. To solve the problem, it is a good idea to make a list of files on the system before you install ucsip-tcp, and one afterwards, and then compares them using the **diff** utility to find out what files were placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep root]# find /* > ucsip-tcp1
  ```
- And the following one after you install the software:
  ```
  [root@deep root]# find /* > ucsip-tcp2
  ```
- Then use the following command to get a list of what changed:
  ```
  [root@deep root]# diff ucsip-tcp1 ucsip-tcp2 > ucsip-tcp-Installed
  ```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. In our example above, we use the /root directory of the system to stock all generated list files.

Compiling - Optimizing & Installing ucsip-tcp

Below are the steps that you must make to configure, compile and optimize the ucsip-tcp software before installing it into your Linux system. First off, we install the program as user 'root' so as to avoid authorization problems.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- This can be done with the following commands:
  ```
  [root@deep /]# cp ucsip-tcp-version.tar.gz /var/tmp/
  [root@deep /]# cd /var/tmp/
  [root@deep tmp]# tar xzpf ucsip-tcp-version.tar.gz
  [root@deep tmp]# cd ucsip-tcp-version
  ```

Step 2
Now, it's important to edit the conf-home file and change the default location where the ucspi-tcp programs will be installed to fit our operating system environment.

- Edit the **conf-home** file (vi conf-home) and change the line:

  ```
  /usr/local
  ```

 To read:

  ```
  /usr
  ```

Step 3
Finally, before going into the compilation of the program, we'll edit the `conf-cc` file and change the default compiler flags to fit our own CPU architecture for better performance.

- Edit the **`conf-cc`** file (`vi conf-cc`) and change the line:

  ```
  gcc -O2
  ```

 To read:

  ```
  gcc -O2 -march=i686 -funroll-loops
  ```

WARNING: Please don't forget to adjust the above optimization `FLAGS` to reflect your own system and `CPU` architecture.

Step 4
Now, we must make a list of files on the system before you install the software and one afterwards then compare them using the **`diff`** utility to find out what files are placed where and finally install `ucspi-tcp` in the system.

```
[root@deep ucspi-tcp-0.88]# make
[root@deep ucspi-tcp-0.88]# cd
[root@deep root]# find /* > ucspi-tcp1
[root@deep root]# cd /var/tmp/ucspi-tcp-0.88/
[root@deep ucspi-tcp-0.88]# make setup check
[root@deep ucspi-tcp-0.88]# chmod 0510 /usr/bin/tcpserver
[root@deep ucspi-tcp-0.88]# chmod 0510 /usr/bin/tcpclient
[root@deep ucspi-tcp-0.88]# cd
[root@deep root]# find /* > ucspi-tcp2
[root@deep root]# diff ucspi-tcp1 ucspi-tcp2 > ucspi-tcp-Installed
```

Step 5
Once the compilation, optimization and installation of the software have been finished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete `ucsip-tcp` and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf ucsip-tcp-version/
[root@deep tmp]# rm -f ucsip-tcp-version.tar.gz
```

The `rm` command as used above will remove all the source files we have used to compile and install `ucsip-tcp`. It will also remove the `ucsip-tcp` compressed archive from the `/var/tmp` directory.

Using ucsip-tcp

As I said previously, ucsip-tcp comes with many small tools to use when you need to run third party programs that cannot start as daemon on your server. Below I show you how to use ucsip-tcp with this software. This is just a general overview since more detailed commands are explained in this book when we need to use ucsip-tcp with specific software.

Here I just explain the options related to security and performance. All examples are based on tcpserver, since it is the one we need to be able to run software via super-server.

The tcpserver program:

The tcpserver is used to accept incoming TCP connections and waits for connections from TCP clients of the third party program we want it to run. We use it as follow:

```
[root@deep /]# tcpserver opts host port prog
```

- ✓ Where "opts" is a series of getopt-style options to use with it.
- ✓ Where "host" is one argument representing the host on which we want it to run.
- ✓ Where "port" is one argument representing the port on which we want it to listen.
- ✓ Where "prog" consists of one or more arguments representing the name of the binary program to run with it.

The biggest part of the tcpserver command is the options that we can use with it. Here are the most interesting to take care of:

The "-c" option is used to define the maximum number of simultaneous connections that we want tcpserver to handle. The default value of this option is "40" meaning no more than 40 simultaneous connections could be handled by tcpserver. For a highly loaded server, it becomes clear that we will need to change the default value for something more adequate. Therefore here is where you can define the value that you need.

The "-g" option is used to define the GID under which we want tcpserver to run the specified program once started. This option becomes useful when we have to run programs under GID's other than the super-user "root" for security reasons. To be able to use this option, you must be sure that the program you are trying to run with tcpserver can run with GID other than "root".

The "-u" option is used to define the UID under which we want tcpserver to run the specified program once started. This option becomes useful when we have to run program under other UID than the super-user "root" for security reasons. To be able to use this option, you must be sure that the program you are trying to run with tcpserver can run with UID other than "root".

The "-D" option is used to inform tcpserver to never delay sending data by enabling TCP_NODELAY. This option is useful to improve performance of the running program with tcpserver. I highly recommend you to use it with your program.

The "-H" option is used to avoid loops on the system by informing tcpserver to not look up the remote host name in DNS. This option is useful to limit possible timeouts. It also improves performance of the program since no look up is performed via DNS. I highly recommend you to use it with your program to speed up connections.

The "-R" option is used to avoid loops on the system by informing tcpserver to not attempt look up the name listed in DNS for the remote host. This option is useful to limit possible timeouts due to misconfigured identd server or unavailable identd server. It also improves performance of the program since no look up is performed at all. I highly recommend you to use it with your program.

The "-l" option is the same as for the above two options but informs tcpserver not to attempt to look up the local host name of the server on which it runs in DNS. Once again, this option is useful to limit possible timeouts and improve performance of the program. I highly recommend you to use it with your program.

Now some real examples to illustrate tcpserver commands:

Example 1: We run vsFTPd a FTP server with tcpserver.

```
[root@deep /]# tcpserver -c 4096 -DRHl localhost 0 21 /usr/sbin/vsftpd
```

The above example will run the "/usr/sbin/vsftpd" binary on port 21 "21" and on all available interfaces on the server "0" with no look up and TCP_NODELAY "-DRHl localhost" for 4096 "-c 4096" simultaneous connections with tcpserver.

Example 2: We run ipop3d a POP3 server with tcpserver.

```
[root@deep /]# tcpserver -c 1024 -DRHl localhost 207.35.78.2 110
/usr/sbin/ipop3d
```

The above example will run the "/usr/sbin/ipop3d" binary on port 110 "110" and on IP address 207.35.78.2 with no look up and TCP_NODELAY "-DRHl localhost" for 1024 "-c 1024" simultaneous connections with tcpserver.

CHAPTER 25

Xinetd

IN THIS CHAPTER

1. Compiling - Optimizing & Installing `Xinetd`
2. Configuring `Xinetd`
3. The `/etc/xinetd.d` directory

Linux Xinetd

Abstract
Xinetd is a secure, powerful and efficient replacement for the old Internet services daemons named inetd and tcp_wrappers. Xinetd can control denial-of-access attacks by providing access control mechanisms for all services based on the address of the remote client that wants to connect to the server as well as the ability to make services available based on time of access, extensive logging, and the ability to bind services to specific interfaces.

But wait, Xinetd is NOT efficient or adequate for all services, especially for services like FTP and SSH. It is far better to run these services as standalone daemons (if possible). Loading services like FTP or SSH, as standalone daemons will eliminate load time and will even reduce swapping since non-library code will be shared. Also, most services that required the super-servers to run have now very good access control mechanisms; therefore, don't think that if you run these services through Xinetd you will necessarily gain additional security.

A few security features of Xinetd are:

- ✓ Provides access control mechanisms.
- ✓ Prevents denial of service attacks.
- ✓ Extensive logging abilities.
- ✓ Offloads services to a remote host.
- ✓ Make services available based on time.
- ✓ Limits on the number of servers that can be started.
- ✓ IPv6 support.

I would like to be clear here before going into discussion about Xinetd. All services that required super-server software to run can use ucspi-tcp. Ucspi-tcp is faster than Xinetd, well written and more secure, therefore I highly recommend you to use it instead of Xinetd. Now for users that still want to go with Xinetd, this is the chapter to read.

These installation instructions assume
Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest Xinetd version number is 2.3.5

The procedures given in this chapter are likely to work on all Linux platforms, but we have only tested it on OpenNA Linux and Red Hat Linux.

Packages
The following is based on information as listed by Xinetd as of 2002/05/28. Please regularly check at http://www.xinetd.org/ for the latest status. We chose to install the required component from source file because it provides the facility to fine tune the installation.

Source code is available from:
Xinetd Homepage: http://www.xinetd.org/
You must be sure to download: xinetd-2.3.5.tar.gz

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all installed files on the system in the eventuality of an update in the future. To solve the problem, it is a good idea to make a list of files on the system before you install Xinetd, and one afterwards, and then compares them using the **diff** utility to find out what files were placed where.

- Simply run the following command before installing the software:
```
[root@deep root]# find /* > Xinetd1
```

- And the following one after you install the software:
```
[root@deep root]# find /* > Xinetd2
```

- Then use the following command to get a list of what changed:
```
[root@deep root]# diff Xinetd1 Xinetd2 > Xinetd-Installed
```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. Related to our example above, we use the /root directory of the system to stock all generated list files.

Compiling - Optimizing & Installing Xinetd

Below are the steps that you must make to configure, compile and optimize the Xinetd software before installing it on your system. First off, we install the program as user 'root' so as to avoid any authorization problems.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- This can be done with the following commands:
```
[root@deep /]# cp xinetd-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf xinetd-version.tar.gz
```

Step 2
After that, move into the newly created Xinetd directory then configure and optimize it.

- To move into the newly created Xinetd directory use the following command:
```
[root@deep tmp]# cd xinetd-2.3.5/
```

- To compile and optimize Xinetd use the following compilation lines:
```
CFLAGS="-O2 -march=i686 -funroll-loops"; export CFLAGS
./configure \
--prefix=/usr \
--sysconfdir=/etc \
--with-loadavg \
--mandir=/usr/share/man
```

Step 3
Now, we must make a list of files on the system before we install the software and one afterwards then compare them using the **diff** utility to find out what files are placed where and finally install Xinetd on the server:

```
[root@deep xinetd-2.3.5]# make
[root@deep xinetd-2.3.5]# cd
[root@deep root]# find /* > Xinetd1
[root@deep root]# cd /var/tmp/xinetd-2.3.5/
[root@deep xinetd-2.3.5]# make install
[root@deep xinetd-2.3.5]# rm -f /usr/sbin/itox
[root@deep xinetd-2.3.5]# rm -f /usr/sbin/xconv.pl
[root@deep xinetd-2.3.5]# rm -f /usr/share/man/man8/itox.8
[root@deep xinetd-2.3.5]# chmod 0510 /usr/sbin/xinetd
[root@deep xinetd-2.3.5]# strip /usr/sbin/xinetd
[root@deep xinetd-2.3.5]# cd
[root@deep root]# find /* > Xinetd2
[root@deep root]# diff Xinetd1 Xinetd2 > Xinetd-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

Step 4
Once the compilation, optimization and installation of the software have been finished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete Xinetd and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf xinetd-version/
[root@deep tmp]# rm -f xinetd-version.tar.gz
```

The rm command as used above will remove all the source files we have used to compile and install Xinetd. It will also remove the Xinetd compressed archive from the /var/tmp directory.

Step 5
One last thing to do is to remove /etc/hosts.allow and /etc/hosts.deny files (if they exist) from your system since we don't need them anymore. Files hosts.allow and hosts.deny are installed by other Linux RPM packages during install. So we can remove them with the following commands.

- To delete hosts.allow and hosts.deny files from your system, use the commands:

```
[root@deep /]# rm -f /etc/hosts.allow
[root@deep /]# rm -f /etc/hosts.deny
```

Configuring Xinetd

After Xinetd has been built and installed successfully in your system, your next step is to configure and customize its configuration files to fit your needs.

- ✓ /etc/xinetd.conf (The Xinetd Configuration File)
- ✓ /etc/init.d/xinetd (The Xinetd Initialization File)

/etc/xinetd.conf: The xinetd Configuration File

The /etc/xinetd.conf file is the main configuration file for Xinetd. It is in this configuration file that Xinetd gets all of its information and the way it should run on your system. It controls the default settings that apply to all services handled by Xinetd.

Step 1
Here, are the most important attributes of the xinetd.conf file for maximum security. Texts in bold are the parts of the configuration file that must be customized and adjusted to meet our needs.

- Create the **xinetd.conf** file (touch /etc/xinetd.conf) and add the following lines. Below is what we recommend you enter:

```
defaults
{
        instances               = 60
        log_type                = SYSLOG authpriv
        log_on_success          = HOST PID
        log_on_failure          = HOST
        only_from               =
        per_source              = 5
}

includedir /etc/xinetd.d
```

This tells the xinetd.conf file to set itself up for this particular configuration with:

instance = 60
This option "instance" is used to specify the maximum number of simultaneous connections allowed for any service that runs through Xinetd. If a specific service does not specify its own instance, that service will be limited to the default value specified with this option. "UNLIMITED" can by used to specify an unlimited number of connections for all services handled by Xinetd. This is a security feature to protect Xinetd from some **D**enial **o**f **S**ervice (DoS) attacks.

log_type = SYSLOG authpriv
This option "log_type" is used to specify the log format to use (you may choose FILE or SYSLOG). For the FILE format, this means the full path to the log file, and for the SYSLOG format, the syslog facility of the system.

log_on_success = HOST PID
This option "log_on_success" is used to specify what we should log when a server is started. This attribute accepts five different values: PID to log the pid Xinetd uses to spawn the server, HOST to logs the remote host's IP address, USERID to logs the UID of the remote user as returned by the remote identd daemon service (if available), EXIT logs the exit status of the server when it exits, and DURATION to logs the duration of the server session.

log_on_failure = HOST
This option "log_on_failure" is used to specify what we should log when the server could not be started for any reason. This attribute accepts four valid values: HOST to logs the remote host's IP address, USERID to logs the UID of the remote user as returned by remote identd daemon service (if available), ATTEMPT to acknowledge that a failed attempt was made, and RECORD grabs as much info as is possible about the remote end.

only_from =
This option "only_from" is used to specify which remote hosts are allowed to connect to the server and use services. By default denying access to every one, is the first step of a reliable security policy. Not giving a value to this option makes every connection fail. This is the same principle as for the IPTABLES Firewall rules. In our example we deny access to all connections then, allow access by means of the same option for specific service under the /etc/xinetd.d directory.

per_source = 5
This option "per_source" is used to specify the maximum number of connections a specific remote IP address can have to a specific local service. It can either be an integer, or the special value "UNLIMITED" for an unlimited number of connections. This attribute will protect from **D**enial **o**f **S**ervice (DoS) attacks.

includedir /etc/xinetd.d
This option "includedir" is used to specify the location of a directory under which all files inside that directory will be parsed as Xinetd configuration files.

Step 2
Now, set the permission mode of the xinetd.conf file to be (0600/-rw-------) and owned by the super-user 'root' for security reasons.

- To change the permission mode and ownership of the xinetd.conf file, use:
```
[root@deep /]# chmod 600 /etc/xinetd.conf
[root@deep /]# chown 0.0 /etc/xinetd.conf
```

The /etc/xinetd.d directory

Now that our xinetd.conf file is configured, we have to create files for services that we expect to run through Xinetd. These files should be created under the /etc/xinetd.d directory because the xinetd.conf file expects to find them under this location.

For each service that we want to run with Xinetd, we have to create a file based on the name of the service and configure it. Below we will show you different configuration options for pop3s, time, chargen, echo, daytime, and imaps services. In this way you will have a good idea of specific parameters available for different services, which can run through Xinetd and how to use them.

If you remember, I said at the beginning of this tutorial that we don't need to install TCP WRAPPER anymore with Xinetd on Linux. TCP WRAPPER is a program that controls who can or cannot log in to the server and from where. Contrary to its predecessor (inetd), Xinetd has two features already built included, which allows you to have the same, and even better, control as the TCP WRAPPER program could offer.

The first feature is called "only_from"; this attribute with its list of IP addresses determines the remote host to which the particular service is available.

The second attribute is named "no_access" and determines the remote hosts to which the particular service is unavailable.

The use of these two options can determine the location access control enforced by Xinetd. One very interesting part of these two attributes is the possibility to build a very restrictive but flexible access control program.

For each service, we must check or change the default one to fit our requirements and operating system. Text in bold are the parts of the configuration file that must be customized and adjusted to satisfy our needs.

/etc/xinetd.d/pop3s: The pop3s configuration file:

The /etc/xinetd.d/pop3s file is the configuration file for pop3s service.

- Create the **pop3s** file (touch /etc/xinetd.d/pop3s) and add the following lines. Below is what we recommend you use for pop3s service with Xinetd:

```
service pop3s
{
        socket_type             = stream
        wait                    = no
        user                    = root
        server                  = /usr/sbin/ipop3d
        only_from               = 0.0.0.0/0
        no_access               = 207.35.78.10
        instances               = 30
        log_on_success          += DURATION HOST
        log_on_failure          += HOST
        nice                    = -2
        disable                 = yes
}
```

This tells the pop3s file to set itself up for this particular configuration with:

service pop3s
This option "service" is used to specify a unique name for the service you wish to configure. This name is what the program uses to look up the service information in the /etc/services file. Be aware that you cannot use any name to set this attribute, protocols exist for this purpose and if you don't know correct name to enable your needed service, then edit the /etc/services file and look inside it for the appropriate name.

socket_type = stream
This option "socket_type" is used to specify the type of socket to be used for the specific service. Available values are: "stream", "dgram", "raw", "rdm", or "seqpacket", depending on whether the socket is a stream, datagram, raw, reliably delivered message, or sequenced packet socket. For pop3s service we must choose and set this attribute to the value "stream".

wait = no
This option "wait" is used to specifies if a datagram server connected to its peer allows the xinetd daemon to receive further messages on the socket or not. If the answer is yes (xinetd can receive further messages on the socket with this program) then this program should use the "nowait" entry and we will set the value of wait to no to indicate the "nowait" entry. This is the default for most services under Xinetd.

`user = root`
This option "`user`" is used to define the user name the server should run as. Usually this value is set to the super-user "`root`" but in some cases, it could be other unprivileged users, therefore it is preferable to verify with the service that you want to run with `Xinetd` if other values are possible for better security. This is a security feature.

`server = /usr/sbin/ipop3d`
This option "`server`" is used to define the pathname of the binary program to run through `Xinetd` when a request is found on its socket.

`only_from = 0.0.0.0/0`
This option "`only_from`" is used to control which remote hosts are allowed to connect to the server and use the service. Remember that we have denied access to everyone in the x`inetd.conf` file; therefore we must allow access for the specific service (`pop3s`) in this file. For a public mail server that runs an `IMAP` or `POP` server it is important to set the value to `0.0.0.0/0` since connections may come from different locations. This is a security feature.

`no_access = 207.35.78.10`
This option "`no_access`" is used to specify which remote hosts are not allowed to connect to the server and use the service. In our example, we don't allow the client with `IP` address of 207.35.78.10 to connect to the `pop3s service`. As you can see, the combination of both attributes (`only_from` and `no_access`) allows us to full control of what can pass through our network. This is a security feature.

`instance = 30`
As noted in the previous `xinetd.conf` file, the option "`instance`" is used to specify the maximum number of requests any service may handle at once. Setting this attribute in the `pop3s` file should override whatever is in the `xinetd.conf` file. This is a performance feature.

`log_on_success += DURATION HOST`
As noted for the `xinetd.conf` file, the option "`log_on_success`" is used to specify what is to be logged when a server is started. For a `pop3s` connection we choose to log the duration of the server session (`DURATION`) and the remote host's `IP` address (`HOST`). Note the assignment operator in this case `'+='` which means to add the value to the set. This is a security feature.

`log_on_failure += HOST`
Again, as in the previous `xinetd.conf` file, the option "`log_on_failure`" is used to specify what is to be logged when the server could not be started. For a `pop3s` connection we choose to log the remote host's `IP` address (`HOST`). This is a security feature.

`nice = -2`
This option "`nice`" is used to modify the default scheduling priority of the process (`pop3s`). The default priority for a process, like `pop3s,` is 10 (range goes from -20 (highest priority) to 19 (lowest)). By increasing the priority of the `pop3s` process the connection time will be faster. This hack can by applied to any other processes running on UNIX; see the manual page about the command `nice (1)` for more information on this feature.

`disable = yes`
This option "`disable`" is used to inform `Xinetd` if the specified service should be enabled or disabled. All service configuration files have this option turned off by default. Therefore you have to change the default option of "`yes`" to "`no`" if you want to enable the specified service with `Xinetd`. If you keep the default setting of "`yes`", then `Xinetd` will NOT start the service.

/etc/xinetd.d/time: The time configuration file:

The /etc/xinetd.d/time file is the configuration file for time service. Here, we'll only explain new options that do not appear in other configuration files.

- Create the **time** file (touch /etc/xinetd.d/time) and add the following lines. Below is what we recommend you to use for time service with Xinetd:

```
service time
{
        socket_type             = stream
        wait                    = no
        user                    = root
        type                    = INTERNAL
        id                      = time-stream
        protocol                = tcp
        only_from               = 207.35.78.0/24 192.168.1.0/24
        no_access               = 207.35.78.10
        disable                 = yes
}

service time-udp
{
        socket_type             = dgram
        wait                    = yes
        user                    = root
        type                    = INTERNAL
        id                      = time-dgram
        protocol                = udp
        only_from               = 207.35.78.0/24 192.168.1.0/24
        no_access               = 207.35.78.10
        port                    = 37
        disable                 = yes
}
```

This tells the time file to set itself up for this particular configuration with:

socket_type = stream and socket_type = dgram
As described previously, the option "socket_type" specifies the type of socket to be used for the specific service. The available values are: "stream", "dgram", "raw", "rdm", or "seqpacket", depending on whether the socket is a stream, datagram, raw, reliably delivered message, or sequenced packet socket. For the time service we must choose "stream" for TCP connection and "dgram" for UDP connection.

wait = no and wait = yes
As described previously, the option "wait" specifies if a datagram server connected to its peer allows the xinetd daemon to receive further messages on the socket or not. If the answer is yes (xinetd can receive further message on the socket with this program) then this program should use the "nowait" entry and we will set the value of wait to no to indicate the "nowait" entry.

It's important to note that UDP protocol by its very nature does not allow peer daemons to receive further messages and it is for this reason that we set the "wait" attribute for UDP version of the time server to yes (xinetd cannot receive further message on the socket with this program).

`type = INTERNAL`
Well, here we see a new attribute; the option "`type`" is used to specify the type of service. The available values are: "`RPC`", "`INTERNAL`", and "`UNLISTED`", depending on whether the specific program is an `RPC` service (`type = RPC`), or a service provided by `Xinetd` (`type = INTERNAL`) or if it is a service not listed in a standard system file like `/etc/rpc` for `RPC` services, or `/etc/services` for non-`RPC` services (`type = UNLISTED`). In our case `time` server is provided by `Xinetd`.

`id = time-stream` and `id = time-dgram`
Ok, here is another new attribute; By default with `Xinetd` the attribute "`id`" is the same as the `service` name, but sometimes (as in our `time` server example) services can use different protocols (`TCP` or `UDP`) and therefore need to be described with separate entries in the configuration file, for `Xinetd` to be able to distinguish them. With this "`id`" attribute, we can uniquely identify service, which uses different communication protocols like `TCP` and `UDP`.

`protocol = tcp` and `protocol = udp`
We continue our discovery with the new attribute called "`protocol`", this option determines the type of protocol that is employed by the specific service. In our example `time` server use both the `TCP` and `UDP` protocols and we specify this with the "`protocol`" attribute.

`port = 37`
Sometimes, and especially with the `UDP` protocol, it is preferable to specify to the program on which port we want the connection to be established. This option "`port`" makes it possible by determining the service port.

`/etc/xinetd.d/chargen`: The `chargen` configuration file:
The `/etc/xinetd.d/chargen` file is the configuration file for `chargen` service.

- Create the **`chargen`** file (`touch /etc/xinetd.d/chargen`) and add the following lines. Below is what we recommend you to use for the `chargen` service with `Xinetd`:

```
service chargen
{
        socket_type             = stream
        wait                    = no
        user                    = root
        type                    = INTERNAL
        id                      = chargen-stream
        protocol                = tcp
        only_from               = 207.35.78.0/24 192.168.1.0/24
        no_access               = 207.35.78.10
        disable                 = yes
}

service chargen-udp
{
        socket_type             = dgram
        wait                    = yes
        user                    = root
        type                    = INTERNAL
        id                      = chargen-dgram
        protocol                = udp
        only_from               = 207.35.78.0/24 192.168.1.0/24
        no_access               = 207.35.78.10
        port                    = 19
        disable                 = yes
}
```

Here, you are supposed to know and understand every attribute shown above. If you have problems, then refer to the previous `time` service configuration parameters for more information.

`/etc/xinetd.d/echo`: The `echo` configuration file:

The `/etc/xinetd.d/echo` file is the configuration file for `echo` service.

- Create the **`echo`** file (`touch /etc/xinetd.d/echo`) and add the following lines. Below is what we recommend you to use for `echo` service with `Xinetd`:

```
service echo
{
        socket_type             = stream
        wait                    = no
        user                    = root
        type                    = INTERNAL
        id                      = echo-stream
        protocol                = tcp
        only_from               = 207.35.78.0/24 192.168.1.0/24
        no_access               = 207.35.78.10
        disable                 = yes
}

service echo-udp
{
        socket_type             = dgram
        wait                    = yes
        user                    = root
        type                    = INTERNAL
        id                      = echo-dgram
        protocol                = udp
        only_from               = 207.35.78.0/24 192.168.1.0/24
        no_access               = 207.35.78.10
        port                    = 7
        disable                 = yes
}
```

`/etc/xinetd.d/daytime`: The `daytime` configuration file:

The `/etc/xinetd.d/daytime` file is the configuration file for `daytime` service.

- Create the **`daytime`** file (`touch /etc/xinetd.d/daytime`) and add the following lines. Below is what we recommend you to use for `daytime` service with `Xinetd`:

```
service daytime
{
        socket_type             = stream
        wait                    = no
        user                    = root
        type                    = INTERNAL
        id                      = daytime-stream
        protocol                = tcp
        only_from               = 207.35.78.0/24 192.168.1.0/24
        no_access               = 207.35.78.10
        disable                 = yes
}
```

```
service daytime-udp
{
        socket_type             = dgram
        wait                    = yes
        user                    = root
        type                    = INTERNAL
        id                      = daytime-dgram
        protocol                = udp
        only_from               = 207.35.78.0/24 192.168.1.0/24
        no_access               = 207.35.78.10
        port                    = 13
        disable                 = yes
}
```

`/etc/xinetd.d/imaps`: The `imaps` configuration file:

At this stage of your reading, you know the most important attributes and values for `Xinetd`, but are aware that many others exist, like the "`redirect`" attribute, which allows a `TCP` service to be redirected to another host in your network. This option is useful when your internal machines are not visible to the outside world and you want to make it visible. The "`bind`" attribute is another one, which allows a service to be bound to a specific interface of your choice on the server for maximum security.

The `/etc/xinetd.d/imaps` file is the configuration file for `imaps` service. Here, we explain only the new options that do not appear in other configuration files.

- Create the **`imaps`** file (`touch /etc/xinetd.d/imaps`) and add the following lines. Below is what we recommend you to use for `imaps` service with `Xinetd`:

```
service imaps
{
        socket_type             = stream
        wait                    = no
        user                    = root
        server                  = /usr/sbin/imapd
        only_from               = 0.0.0.0/0
        no_access               = 207.35.78.10
        instances               = 30
        log_on_success          += DURATION HOST
        log_on_failure          += HOST
        nice                    = -2
        redirect                = 192.168.1.14 993
        bind                    = 207.35.78.3
        disable                 = yes
}
```

This tells the `imaps` file to set itself up for this particular configuration with:

`redirect = 192.168.1.14 993`
This attribute "`redirect`" allows a `TCP` service received on the specified port (in our example the port 993) to be redirected to another host (`192.168.1.14`) by forwarding all data between the two hosts.

`bind = 207.35.78.3`
This attribute "`bind`" allows a service of your choice to be bound to a specific interface on the server. In our case the `imaps` service is bound to the interface `207.35.78.3`. Therefore, if someone from an allowed host tries to bind to another interface on the server, then `Xinetd` will refuse the connection. This is a security feature.

/etc/init.d/xinetd: The Xinetd Initialization File

The /etc/init.d/xinetd script file is responsible for automatically starting and stopping the Xinetd server on your Linux system. Please note that the following script is suitable for Linux operating systems that use SystemV. If you Linux system use some other methods like BSD, you'll have to adjust the script below to make it work for you.

Step 1
Create the **xinetd** file (touch /etc/init.d/xinetd) and add the following lines:

```
#!/bin/bash

# This shell script takes care of starting and stopping Xinetd.
#
# chkconfig: 345 56 50
# description: Xinetd is a powerful replacement for inetd. It has access \
#               control machanisms, extensive logging capabilities, the \
#               ability to make services available based on time, and can \
#               place limits on the number of servers that can be started, \
#               among other things.
#
# processname: /usr/sbin/xinetd
# config: /etc/sysconfig/network
# config: /etc/xinetd.conf
# pidfile: /var/run/xinetd.pid

prog="Xinetd"
PATH=/sbin:/bin:/usr/bin:/usr/sbin

# Source function library.
. /etc/init.d/functions

# Get config.
test -f /etc/sysconfig/network && . /etc/sysconfig/network
test -f /etc/sysconfig/xinetd && . /etc/sysconfig/xinetd

# Check that networking is up.
[ ${NETWORKING} = "yes" ] || exit 0

[ -f /usr/sbin/xinetd ] || exit 1
[ -f /etc/xinetd.conf ] || exit 1

RETVAL=0

start() {
        echo -n $"Starting $prog: "
        LANG=en_US
        LC_TIME=en_US
        LC_ALL=en_US
        LC_MESSAGES=en_US
        LC_NUMERIC=en_US
        LC_MONETARY=en_US
        LC_COLLATE=en_US
        export LANG LC_TIME LC_ALL LC_MESSAGES LC_NUMERIC LC_MONETARY
LC_COLLATE
        unset HOME MAIL USER USERNAME

        daemon xinetd -stayalive -reuse -pidfile /var/run/xinetd.pid
"$EXTRAOPTIONS"
        RETVAL=$?
        echo
        touch /var/lock/subsys/xinetd
```

```
        return $RETVAL
}

stop() {
        echo -n $"Stopping $prog: "
        killproc xinetd
        RETVAL=$?
        echo
        rm -f /var/lock/subsys/xinetd
        return $RETVAL

}

reload() {
        echo -n $"Reloading configuration: "
        killproc xinetd -USR2
        RETVAL=$?
        echo
        return $RETVAL
}

restart() {
        stop
        start
}

condrestart() {
        [ -e /var/lock/subsys/xinetd ] && restart
        return 0
}


# See how we were called.
case "$1" in
    start)
        start
        ;;
    stop)
        stop
        ;;
    restart)
        restart
        ;;
    reload)
        reload
        ;;
    condrestart)
        condrestart
        ;;
    *)
        echo $"Usage: $0 {start|stop|restart|condrestart|reload}"
        RETVAL=1
esac
exit $RETVAL
```

Step 2
Once the xinetd script file has been created, it is important to make it executable, change its default permissions, create the necessary links and start it. Making this file executable will allow the system to run it, changing its default permission to allow only the root user to change this file for security reasons, and the creation of symbolic links will let the processes that control the initialization of Linux, which is in charge of starting all the normal and authorized processes that need to run at boot time on your system, to start the program automatically for you at each boot.

- To make this script executable and to change its default permissions, use the command:
```
[root@deep /]# chmod 700 /etc/init.d/xinetd
[root@deep /]# chown 0.0 /etc/init.d/xinetd
```

- To create the symbolic rc.d links for Xinetd, use the following command:
```
[root@deep /]# chkconfig --add xinetd
[root@deep /]# chkconfig --level 345 xinetd on
```

- To start Xinetd software manually, use the following command:
```
[root@deep /]# /etc/init.d/xinetd start
Starting Xinetd:          [OK]
```

Further documentation

For more details, there are some manual pages about Xinetd that you could read:

```
$ man xinetd.conf (5)        - Configuration settings for Xinetd.
$ man xinetd.log (8)         - Xinetd service log format.
$ man xinetd (8)             - The extended Internet services daemon.
```



Part VII

Management & Limitation

CHAPTER 26

NTP

IN THIS CHAPTER

1. Compiling - Optimizing & Installing `NTP`
2. Configuring `NTP`
3. Running `NTP` in Client Mode
4. Running `NTP` in Server Mode
5. Running `NTP` in a chroot jail
6. `NTP` Administrative Tools

Linux NTP

Abstract

Having all computers inside our network environment synchronized is a very important part of security measure. This allow us to get accurate information of different report we may have to read as well as having all servers reporting all networking messages and services in the same time. This also can improve performance of your entire network because all systems are synchronized together.

A lot of services rely on accurate time with Linux to properly function, we can just take as example the DNS protocol that heavily rely on synchronized time between both DNS servers to make a zone transfer. Other examples exist like the web server to report real time static and web information to the users. In general, all services need to have an accurate time to properly report different information to the administration and correctly function on the network. Therefore we cannot avoid installing a time server on our networking area if we want to participate in the new age of computer security.

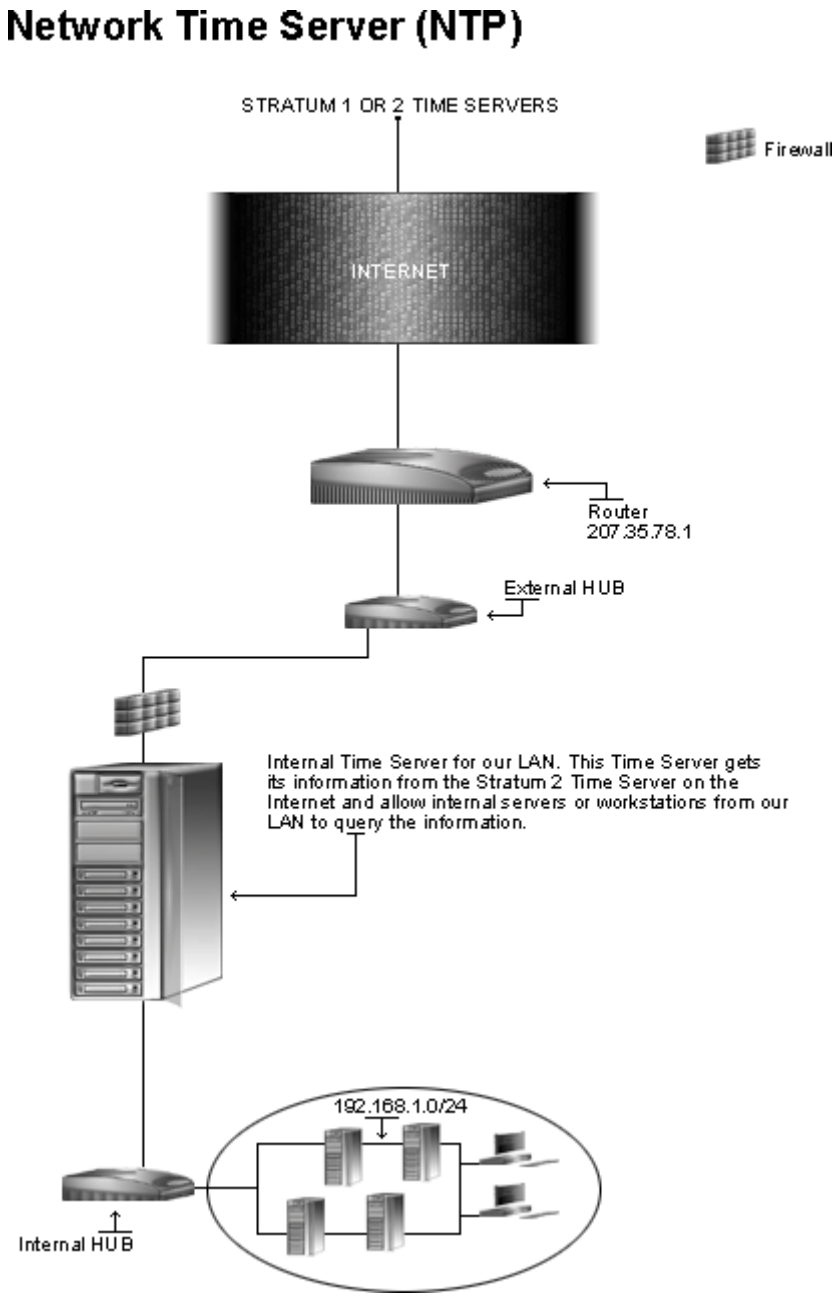
The **N**etwork **T**ime **P**rotocol (NTP) is used to synchronize a computer's time with another reference time source. NTP contains utilities and daemons that will synchronize your computers time to **C**oordinated **U**niversal **T**ime (UTC) via the NTP protocol and NTP servers.

The **N**etwork **T**ime **P**rotocol (NTP) defines a set of procedures for synchronizing clocks on hosts connected to a network with access to the Internet. NTP is based on a multi tiered system where each layer is called a stratum. Servers at the top or in stratum 1 are directly connected to atomic clocks or radio based time receivers which are special hardware made for this purpose. By compensating for their distance from the authoritative time sources these, receivers provide highly accurate time services to stratum 1 servers which again provide accurate time services to stratum 2 servers, etc.

An important factor in getting a network correctly set up with NTP is the selection of servers from which time is obtained. Depending on your network size you will be using either public stratum 1 or 2 time servers or may create your own private or public stratum 1 time server with the appropriate receiving device. In most case we only need to use stratum 2 to avoid to highly load stratum 1 time servers or better configure one of our server as stratum 1 time server for our entire network and machines. This is a good solution for best security with NTP.

The NTP software package includes an ntptrace utility that gives the offset and network distance of NTP servers as well as their parent time servers. Finding the best servers was much more time consuming than installing the software and this is where you should concentrate most of your energy in this chapter.

It is recommended to firstly find the best time servers which are not too far from your physical location and make arrangement with the administrator of these time server to have authorization to use them. Of course most time server are open time server and you are free to use them as you want but it is preferable and polite to advice the administrator of your intention to use their time servers to synchronize your network time.



This is a graphical representation of the `NTP` configuration we use in this book. Please note that lot possibilities exist, and depend of your needs, and network architecture design.

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, at personal discretion).
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest NTP version number is 4.1.1a

The procedures given in this chapter are likely to work on all Linux platforms, but we have only tested it on OpenNA Linux and Red Hat Linux.

Packages

The following is based on information listed by NTP as of 2002/03/28. Please regularly check http://www.ntp.org/ for the latest status. We chose to install the required component from a source file because it provides the facility to fine tune the installation.

Source code is available from:
NTP Homepage: http://www.ntp.org/
NTP FTP site: 128.4.40.10
You must be sure to download: ntp-4.1.1a.tar.gz

Prerequisites

NTP requires that the software below is already installed on your system to be able to compile successfully. If this is not the case, you must install it. Please make sure you have this program installed on your machine before you proceed with this chapter.

- ✓ libcap is required to run NTP on your system.
- ✓ libcap-devel required to build NTP on your system.

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all the files installed on the system in the eventuality of an update in the future. To solve the problem, it is a good idea to make a list of files on the system before you install NTP, and then one afterwards, and then compare them using the **diff** utility to find out what files were placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep root]# find /* > NTP1
  ```
- And the following one after you install the software:
  ```
  [root@deep root]# find /* > NTP2
  ```
- Then use the following command to get a list of what changed:
  ```
  [root@deep root]# diff NTP1 NTP2 > NTP-Installed
  ```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. In the example above, we use the /root directory of the system to store all generated list files.

Compiling - Optimizing & Installing NTP

Below are the steps that you must make to configure, compile and optimize the NTP software before installing it onto your system. First off, we install the program as the user 'root' so as to avoid permissioning problems.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- This can be done with the following commands:

```
[root@deep /]# cp ntp-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf ntp-version.tar.gz
```

Step 2
NTP needs a UID and GID to properly run on the system but this UID/GID cannot run as super-user root; for this reason we must create a special user with no shell privileges on the system for running NTP daemon.

- To create this special NTP user on OpenNA Linux, use the following command:

```
[root@deep tmp]# groupadd -g 38 ntp > /dev/null 2>&1 || :
[root@deep tmp]# useradd -c "NTP Server" -d /etc/ntp -g 38 -s /bin/false
-u 38 ntp > /dev/null 2>&1 || :
```

- To create this special NTP user on Red Hat Linux, use the following command:

```
[root@deep tmp]# groupadd -g 38 ntp > /dev/null 2>&1 || :
[root@deep tmp]# useradd -u 38 -g 38 -s /bin/false -M -r -d /etc/ntp ntp
> /dev/null 2>&1 || :
```

The above command will create a null account, with no password, no valid shell, no files owned-nothing but a UID and a GID for the program. Remember that NTP daemon does not need to have a shell account on the server.

Step 3
Now, edit the **shells** file (vi /etc/shells) and add a non-existent shell name "/bin/false", which is the one we used in the useradd command above.

```
[root@deep tmp]# vi /etc/shells
/bin/bash2
/bin/bash
/bin/sh
/bin/false ← This is our added no-existent shell
```

Making NTP to run in chroot jail:

There is an external patch available for NTP that allow us to compile it with chroot support. If you are interested to compile NTP to support and run in chroot jail mode, then I recommend you to follow these steps. If you don't want to compile NTP with chroot jail support, you can simply skip these steps and go directly to next section where we will compile the software for our system. I highly recommend you to compile NTP with chroot support if you want to run this software with more security on your server.

Step 1
Patching NTP to run in chroot jail mode required modifying most of its source codes and the patch is too big to be listed in this documentation. Therefore, we have to retrieve the patch from the OpenNA website available from the following location:
http://www.openna.com/products/books/securing-optimizing-linux/3rdedition/index.htm

Step 2
Once you have a copy of this patch, you should move it under the /var/tmp directory and patch your NTP source files.

- This can be done with the following commands:

```
[root@deep /]# mv ntp-chroot.patch /var/tmp/
[root@deep /]# cd /var/tmp/ntp-4.1.1a/
[root@deep ntp-4.1.1a]# patch –p1 < ../ntp-chroot.patch
```

Compiling NTP:

Once the required modification has been made into the source file of NTP, it is time configure, compile and optimize it for our system.

- To configure and optimize NTP use the following compilation lines:

```
CFLAGS="-O2 -march=i686 -funroll-loops"; export CFLAGS
./configure \
--prefix=/usr \
--bindir=/usr/sbin \
--sbindir=/usr/sbin \
--sysconfdir=/etc \
--localstatedir=/var \
--mandir=/usr/share/man \
--disable-debugging \
--enable-all-clocks \
--enable-parse-clocks
```

Step 1
At this stage the program is ready to be built and installed. We build NTP with the 'make' command and produce a list of files on the system before we install the software, and one afterwards, then compare them using the **diff** utility to find out what files were placed where and finally install NTP.

```
[root@deep ntp-4.1.1a]# make
[root@deep ntp-4.1.1a]# cd
[root@deep root]# find /* > NTP1
[root@deep root]# cd /var/tmp/ntp-4.1.1a/
[root@deep ntp-4.1.1a]# make install
[root@deep ntp-4.1.1a]# strip /usr/sbin/ntp*
[root@deep ntp-4.1.1a]# cd
[root@deep root]# find /* > NTP2
[root@deep root]# diff NTP1 NTP2 > NTP-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

Step 2
Once the compilation, optimization and installation of the software has completed, we can free up some disk space by deleting the program tar archive and the related source directory, since they are no longer needed.

- To delete NTP and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf ntp-version/
[root@deep tmp]# rm -f ntp-version.tar.gz
```

The rm command as used above will remove all the source files we have used to compile and install NTP. It will also remove the NTP compressed archive from the /var/tmp directory.

Configuring NTP

After NTP has been built and installed successfully on your system, the next step is to configure and customize its configuration files to fit your needs.

- ✓ /etc/ntp.conf: (The NTP Configuration File)
- ✓ /etc/ntp.drift: (The Drift File)
- ✓ /etc/sysconfig/ntpd: (The NTP System Configuration File)
- ✓ /etc/init.d/ntpd: (The NTP Initialization File)

The Time Synchronization Hierarchy:

With NTP, each daemon can be a client, server, or peer for other NTP daemons:

- ✓ As client it queries the reference time from one or more servers.
- ✓ As server it makes its own time available as reference time for other clients.
- ✓ As peer it compares its system time to other peers until all the peers finally agree about the "true" time to synchronize to.

These features can be used or mixed to provide hierarchical time synchronization structures which are called stratum levels. A smaller stratum number means a higher level in the hierarchy structure. On top of the hierarchy there is the daemon which has the most accurate time and therefore the smallest stratum number.

Running NTP in Client Mode

This section applies only if you chose to install and use NTP in Client Mode in your system. Client Mode configuration are servers that poll other hosts to get the current time. A Client Mode Time Server can look up times inside and outside your network.

The difference is that when NTP is configured to run in Client Mode, it queries the reference time from one or more servers. A Client Time Server should be run on any systems which are not a Server or Peer Time Server. This is why I begging the configuration of NTP with the Client Mode configuration.

/etc/ntp.conf: The NTP Configuration File

The /etc/ntp.conf file is the main configuration file for NTP. It is in this configuration file that NTP gets all of its configuration information and the way we want it to run. You should use this configuration file for all servers on your network that don't act as a Server or Peer Time Server. To configure a host in Client Mode, there must be a server statement in its NTP configuration file which specifies the name or IP address of each time server to be polled.

Step 1
With this configuration for a Client Mode Time Server, all time synchronizations are queries from a time server outside or inside your network. Text in bold is the parts of the configuration file that must be customized and adjusted to satisfy our needs.

- Create the **ntp.conf** file (touch /etc/ntp.conf) and add the following lines in the file. Below is what I recommend you set.

```
restrict default notrust nomodify ignore
restrict 127.0.0.1
restrict 195.83.132.135 mask 255.255.255.255 nomodify notrap noquery
server 195.83.132.135
server 127.127.1.0
fudge 127.127.1.0 stratum 10
driftfile /etc/ntp.drift
broadcastdelay 0.008
```

This tells the ntp.conf file to set itself up for this particular configuration with:

`restrict default notrust nomodify ignore`
The restrict declaration is an Access Control Commands used to control access to this service. Remember that NTP can be configured to run in different mode of operation and depending of the type of NTP server that we want to configure, we have to allow or deny some access on the time server.

With the above declaration, we prohibit general access to this service. The flags associated with the entry are specified textually. For example, the "notrust" flag indicates that hosts matching this entry, while treated normally in other respects, shouldn't be trusted to provide synchronization even if otherwise so enabled. The "nomodify" flag indicates that hosts matching this entry should not be allowed to do run-time configuration and finally the "ignore" flag indicate to ignore all packets from hosts which match this entry. If this flag is specified neither queries nor time server polls will be responded to.

Therefore, the above declaration with the specified flags means that by default, we don't trust and don't allow any modifications on this Time Server configuration. This is a security feature.

`restrict 127.0.0.1`
The second declaration here means that the local address (127.0.0.1) is unrestricted and we permit all access over the loopback interface. This is a security feature.

`restrict 195.83.132.135 mask 255.255.255.255 nomodify notrap noquery`
`server 195.83.132.135`
Here we define our Time Server source or if you prefer the remote Time Server from which we will query time synchronization for this Client Mode server machine. As you can see, we always use the "restrict" declaration to control access and security of our Client Time Server. In the above declarations, we permit time synchronization with our time source, but do not permit the source to query or modify the service on this system. This is a security feature.

The IP address of the remote Time Server (195.83.132.135) that I use here as an example is a real working Time Server from the laboratory for analysis and architecture of systems from France. This Time Server is an Open Time Server that everyone could use to synchronize their Client Time machine but I highly recommend you to find a Time Server closed to your geographical location and use it instead of the above example. Also, be kindly and inform the administrator of the remote Time Server for authorization before connecting your systems to their time server. It is good manners to request permission to access a time server by sending e-mail to its administrator.

NOTE: Public active NTP Secondary (stratum 2) Time Servers are available at the following URL: http://www.eecis.udel.edu/~mills/ntp/clock2.htm

```
server 127.127.1.0
fudge 127.127.1.0 stratum 10
```

Here we define a fake driver intended for backup and when no outside source of synchronized time is available. This driver is never used for synchronization, unless no other synchronization source is available. It is useful to have the above lines defined inside our configuration file because this provides some more robustness in case something goes wrong with the software.

Take a note to the pseudo IP address "127.127.1.0". This IP address must not be mixed up with "127.0.0.1", which is the IP of the localhost on the system. NTP uses this pseudo IP address also called the local clock to access its own system clock.

```
driftfile /etc/ntp.drift
broadcastdelay 0.008
```

When the NTP daemon starts for the first time it compute possible error in the intrinsic frequency of the clock on the computer it is running on. This frequency error usually takes about a day or so after the daemon is started to compute a good estimate needed to synchronize closely to its server. Once the initial value is computed, it will change only by relatively small amounts during the course of continued operation.

The "driftfile" declaration is used to specify to the daemon the name of a file where it may store the current value of the frequency error so that, if the daemon is stopped and restarted, it can reinitialize itself to the previous estimate and avoid the day's worth of time it will take to re-compute the frequency estimate.

Step 2
Now, set the permission mode of the ntp.conf file to be (0644/-rw-r--r--) and owned by the user 'root' with group permission set to "root" user for security reason.

- To change the permission mode and ownership of ntp.conf file, use:

```
[root@deep /]# chmod 644 /etc/ntp.conf
[root@deep /]# chown 0.0 /etc/ntp.conf
```

/etc/ntp.drift: The Drift File

This section applies for all type of Time Server (Client, Server or Peer) that you may want to install in your system. When the NTP server is first run on a computer, it is very active in talking to the servers from which it obtains its time so it can determine the network delay and a reasonable starting offset.

It also starts to calculate the local computers drift (the amount by which the clock is fast or slow). After the drift is calculated the normal behavior is to save it in a drift file so that following computer or server restarts it doesn't need to repeat all the work it does the first time it runs.

Step 1
We have to create this file on our server and set the correct permission mode. In the command below, we add "0.0" as a value for this file to starts. The value "0.0" means 0 drift because we don't have any idea of the real local computer's drift. NTP will automatically recalculate this value each hour; therefore we can safety start with "0.0".

- To create the drift file, use the following command:

```
[root@deep /]# echo '0.0' > /etc/ntp.drift
```

Step 2
Now, set the permission mode of the drift file to be (0600/-rw-------) and owned by the user "ntp" with group permission set to "ntp" user for security reason.

- To change the permission mode and ownership of drift file, use:

```
[root@deep /]# chmod 600 /etc/ntp.drift
[root@deep /]# chown ntp.ntp /etc/ntp.drift
```

/etc/sysconfig/ntpd: The NTP System Configuration File

This section applies for all type of Time Server (Client, Server or Peer) that you may want to install in your system. The /etc/sysconfig/ntpd file is used to specify NTP system configuration information, such as if NTP should run in a chroot environment, and if additional options are required to be passed to ntpd daemon at startup.

- Create the **ntpd** file (touch /etc/sysconfig/ntpd) and add the following lines:

```
# This option will run ntpd in a chroot environment.
#
#ROOTDIR="-T /chroot/ntpd"

# Drop root to ID 'ntp:ntp' by default.
OPTIONS="-U ntp"
```

The ROOTDIR="-T /chroot/ntpd" option instructs NTP where the chroot directory is located. Therefore the ntpd daemon reads this line in the /etc/sysconfig/ntpd file and chroot's to the specified directory before starting. Please read the section related to NTP in chroot jail before uncomment the above line.

The "OPTIONS" parameter is used to define the UID under which we want to run NTP. It's important to run NTP under an unprivileged UID for added security. Here we define the UID we want to use to run the ntpd daemon on the server.

Initializing the System Clock

Before starting our ntpd daemon on the server, we must initialize the system clock with the "ntpdate -b <server>" command to synchronize your system's time. It's a good idea to run this command the first time your install NTP on your system and especially when you have any significant deviation of your system's time from the actual time. For example, if your clock is one hour late from the real time, NTP will refuse to start and this is the reason why we should use the "ntpdate" command to solve the problem.

The "ntpdate" command will make a remote connection to the Time Server you have chosen as your primary Time Server and will synchronize your local time with the time of the remote computer. In this way, NTP can be started without problem and will adjust its time by synchronizing every millisecond, etc.

- To initialize your system clock, use the following command:

```
[root@deep /]# ntpdate -b 195.83.132.135
20 Jun 20:43:02 ntpdate[15193]: step time server 195.83.132.135 offset
0.000278 sec
```

Where <195.83.132.135> is one of the available Time Servers in your ntpd.conf file.

/etc/init.d/ntpd: The NTP Initialization File

This section applies for all type of Time Server (Client, Server or Peer) that you may want to install in your system. The /etc/init.d/ntpd script file is responsible for automatically starting and stopping the NTP server. Loading the ntpd daemon as a standalone daemon will eliminate load time and will even reduce swapping since non-library code will be shared.

Please note that the following script is only suitable for Linux OS that use System V. If your Linux system uses some other method, like BSD, you'll have to adjust the script below to make it work for you.

Step 1
Create the **ntpd** script file (touch /etc/init.d/ntpd) and add the following lines:

```
#!/bin/bash

# This shell script takes care of starting and stopping ntpd (NTPv4 daemon).
#
# chkconfig: 345 58 74
# description: NTPD is the NTPv4 daemon that is used to provide time server.

# Source function library.
. /etc/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Source for additional options if we have them.
if [ -f /etc/sysconfig/ntpd ];then
        . /etc/sysconfig/ntpd
fi

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

# If NTPD is not available stop now.
[ -f /usr/sbin/ntpd ] || exit 0
```

```
[ -f /etc/ntp.conf ] || exit 0

# Path to the NTPD binary.
ntpd=/usr/sbin/ntpd

RETVAL=0
prog="NTPD"

start() {
        echo -n $"Starting $prog: "
        daemon $ntpd $OPTIONS
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/ntpd
        return $RETVAL
}

stop() {
        echo -n $"Shutting down $prog: "
        killproc $ntpd
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/ntpd
        return $RETVAL
}

# See how we were called.
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  status)
        status $ntpd
        RETVAL=$?
        ;;
  restart)
        stop
        start
        RETVAL=$?
        ;;
  condrestart)
        if [ -f /var/lock/subsys/ntpd ]; then
            stop
            start
            RETVAL=$?
        fi
        ;;
  *)
        echo $"Usage: $0 {start|stop|status|restart|condrestart}"
        exit 1
esac
exit $RETVAL
```

Step 2
Once the ntpd script file has been created, it is important to make it executable, change its default permissions, create the necessary links and then start it. Making this file executable will allow the system to run it, changing its default permission to allow only the root user to change it for security reasons, and the creation of the symbolic links will let the process control initialization of Linux, which is in charge of starting all the normal and authorized processes that need to run at boot time on your system, start the program automatically for you at each system reboot.

- To make this script executable and to change its default permissions, use the commands:
```
[root@deep /]# chmod 700 /etc/init.d/ntpd
[root@deep /]# chown 0.0 /etc/init.d/ntpd
```

- To create the symbolic rc.d links for NTP, use the following commands:
```
[root@deep /]# chkconfig --add ntpd
[root@deep /]# chkconfig --level 345 ntpd on
```

- To start NTP software manually, use the following command:
```
[root@deep /]# /etc/init.d/ntpd start
Starting NTPD:            [OK]
```

Running NTP in Server Mode

This section applies only if you chose to install and use NTP in Server Mode in your system. When configured in Server Mode operation, NTP become a Time Server which allows it to be polled by a host (the client) that wishes to synchronize with it. You may decide to create your own private or public stratum 1 Time Server with the appropriate receiving device. This is useful and a good practice when you have lot of servers in your network because you can configure one of your systems in Server Mode Time Server for all of your other Client Time Server and improve security of your entire Time Server machines.

/etc/ntp.conf: The NTP Configuration File

Use this ntp.conf configuration file for the server on your network that acts as Server Mode Time Server. In this configuration, we setup the NTP software to get its Time synchronization from remote public Time Servers on the Internet and distribute the information to any servers inside our LAN that is configured to ask it for the time information.

In Server Mode operation, it is highly recommended to define at least 3 external Time Server into the configuration file. This is recommended for redundancy and scalability as well as to get extremely precise accurate time information for our Time Server. In the configuration file below, I use three external Open Time Servers. This is just an example, and it’s to you to find the best Time Servers for your network.

Step 1
To do this, add/change the following lines to your /etc/ntp.conf file. Text in bold is the parts of the configuration file that change from the previous ntp.conf file. Don't forget to adjust the values to satisfy your needs.

- Create the **ntp.conf** file (touch /etc/ntp.conf) and add the following lines in the file. Below is what I recommend you set.

```
restrict default notrust nomodify ignore
restrict 127.0.0.1
restrict 207.35.78.0 mask 255.255.255.0 notrust nomodify notrap
restrict 195.83.132.135 mask 255.255.255.255 nomodify notrap noquery
restrict 216.27.190.202 mask 255.255.255.255 nomodify notrap noquery
restrict 199.212.17.34 mask 255.255.255.255 nomodify notrap noquery
server 195.83.132.135 prefer
server 216.27.190.202
server 199.212.17.34
server 127.127.1.0
fudge 127.127.1.0 stratum 10
driftfile /etc/ntp.drift
broadcastdelay 0.008
```

This tells the ntp.conf file to set itself up for this particular configuration with:

restrict 207.35.78.0 mask 255.255.255.0 notrust nomodify notrap
The above parameter permits systems on this network (207.35.78.0) to synchronize with this time service and do not permit those systems to modify the configuration of this service or use those systems as peers for synchronization. In this way, we limit servers who may ask or query our Time Server for time synchronization to our network range as shown above. This is a security feature and the only one parameter that permit NTP to run in Server Mode.

server 195.83.132.135 prefer
Here we inform NTP that Time Server with IP address (195.83.132.135) is our preferred Time Server to connect to get time information. This is possible by specifying the "prefer" line at the end of the parameter line as shown above. In this way, we can have more than one Time Server reference and define which one should be taken as preference compared to other.

restrict 216.27.190.202 mask 255.255.255.255 nomodify notrap noquery
restrict 199.212.17.34 mask 255.255.255.255 nomodify notrap noquery
server 216.27.190.202
server 199.212.17.34
Here we define two supplemental Time Servers as a backup in case of the first Time Server has some problem. In this way we have a second Time Server defined to continue to retrieve time information. As you can see, we protect our server with the "restrict" declaration before defining the Time Server IP address. You can have more than one Time Server configured in your configuration file. In general, most of us will define three different Time Servers in their configuration.

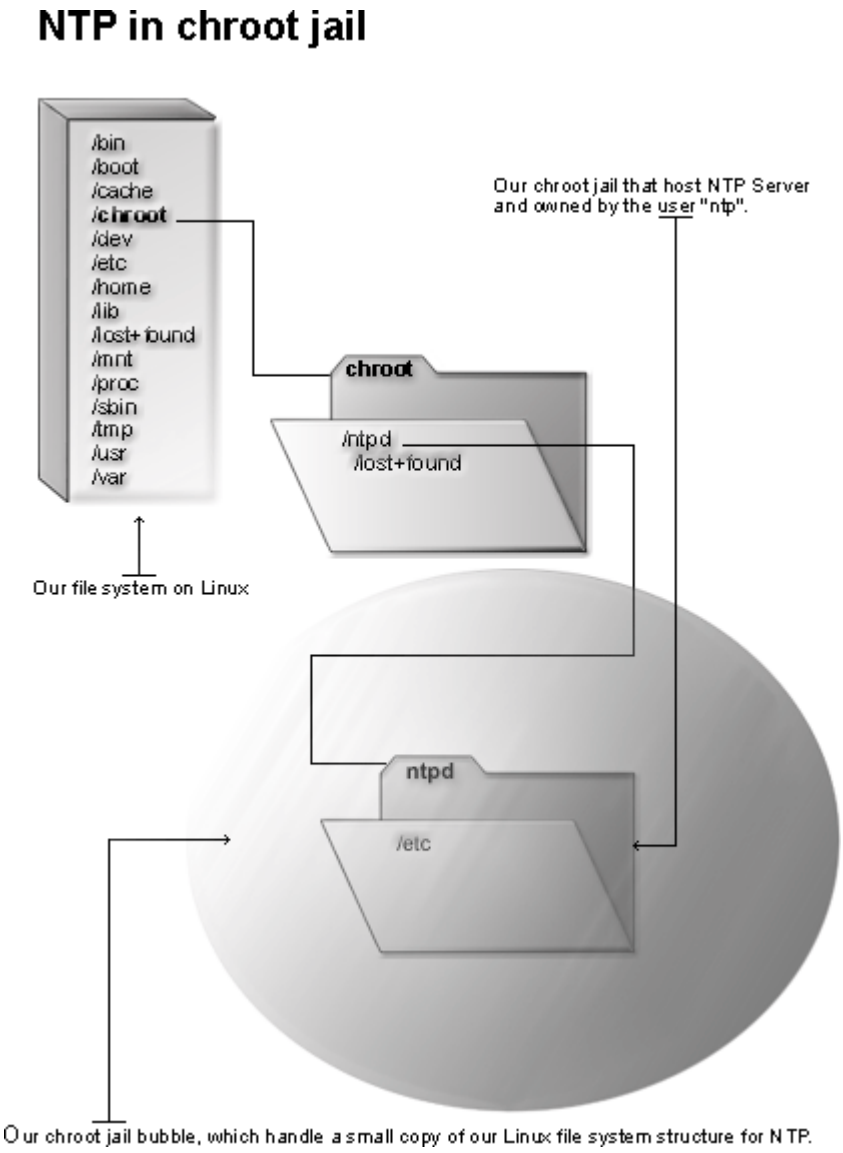
To recap, the above configuration is used on server that is configured to run in Time Server Mode for our LAN. This Time Server get its time information from remote Time Servers on the Internet (195.83.132.135), (216.27.190.202), (199.212.17.34) and allow only all nodes from the 207.35.78.0 network to query it for time information and synchronization.

Step 2
Now, set the permission mode of the `ntp.conf` file to be (`0644/-rw-------`) and owned by the user "`root`" with group permission set to "`root`" user for security reason.

- To change the permission mode and ownership of `ntpd.conf` file, use:
```
[root@deep /]# chmod 644 /etc/ntp.conf
[root@deep /]# chown 0.0 /etc/ntp.conf
```

Running NTP in a chroot jail

This part focuses on preventing `NTP` from being used as a point of break-in to the system hosting it. To minimize this risk, `NTP` can be configured to run in chroot jail environment. The main benefit of a chroot jail is that the jail will limit the portion of the file system the `NTP` daemon program can see to the root directory of the jail. Additionally, since the jail only needs to support `NTP`, the programs related to `NTP` available in the jail can be extremely limited. Most importantly, there is no need for setuid-root programs, which can be used to gain root access and break out of the jail.



Necessary steps to run NTP software in a chroot jail:

What you're essentially doing is creating a skeleton root file system with enough components necessary (directories, files, etc.) to allow Unix to do a chroot when the NTP daemon starts. As you will see further down, running NTP in chroot jail is really easy to accomplish when properly patched with the chroot patch file.

Step 1

The first step to do for running NTP in a chroot jail will be to set up the chroot environment, and create the root directory of the jail. We've chosen /chroot/ntpd for this purpose because we want to put this on its own separate file system to prevent file system attacks. Early in our Linux installation procedure we created a special partition /chroot for this exact purpose.

```
[root@deep /]# /etc/init.d/ntpd stop ← Only if ntpd daemon already run.
Shutting down NTPD:          [OK]

[root@deep /]# mkdir -p /chroot/ntpd/etc
[root@deep /]# chown -R ntp.ntp /chroot/ntpd/etc
```

We need all of the above directories because, from the point of the chroot, we're sitting at "/" and anything above this directory is inaccessible.

Step 2

After that, we must move the main configuration files of NTP into the appropriate places in the chroot jail. This includes the ntp.conf file and all related files as well as the resolv.conf and localtime files.

```
[root@deep /]# mv /etc/ntp.conf /chroot/ntpd/etc/
[root@deep /]# mv /etc/ntp.drift /chroot/ntpd/etc/
[root@deep /]# cp /etc/resolv.conf /chroot/ntpd/etc/
[root@deep /]# cp /etc/localtime /chroot/ntpd/etc/
```

Step 3

For additional security, we can 'chattr' the ntp.conf file in the chroot jail directory.

- This procedure can be accomplished with the following commands:

```
[root@deep named]# cd /chroot/ntpd/etc/
[root@deep etc]# chattr +i ntp.conf
```

WARNING: Don't forget to remove the immutable bit on this file if you have to make some modifications to it later, use the command "chattr -i".

Step 4
At this point, we have to instruct NTP to start in the chroot jail environment. This is done by modifying our original /etc/sysconfig/ntpd and /etc/init.d/ntpd script files. We start with our ntpd file under the /etc/sysconfig directory and continue with our /etc/init.d/ntpd initialization script file.

- Edit the **ntpd** file (vi /etc/sysconfig/ntpd) and add/change the following lines:

```
# This option will run ntpd in a chroot environment.
#
ROOTDIR="-T /chroot/ntpd"

# Drop root to ID 'ntp:ntp' by default.
OPTIONS="-U ntp"
```

The ROOTDIR="-T /chroot/ntpd" option instructs NTP where the chroot directory is located. Therefore the ntpd daemon reads this line in the /etc/sysconfig/ntpd file and chroot's to the specified directory before starting.

- Edit the **ntpd** file (vi /etc/init.d/ntpd) and add/change the following lines:

```
#!/bin/bash

# This shell script takes care of starting and stopping ntpd.
#
# chkconfig: 345 58 74
# description: NTPD is used to provide time server.

# Source function library.
. /etc/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Source for additional options if we have them.
if [ -f /etc/sysconfig/ntpd ];then
        . /etc/sysconfig/ntpd
fi

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

# If NTPD is not available stop now.
[ -f /usr/sbin/ntpd ] || exit 0
[ -f /chroot/ntpd/etc/ntp.conf ] || exit 0

# Path to the NTPD binary.
ntpd=/usr/sbin/ntpd

RETVAL=0
prog="NTPD"

start() {
        echo -n $"Starting $prog: "
        daemon $ntpd $ROOTDIR $OPTIONS
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/ntpd
        return $RETVAL
}
```

```
stop() {
        echo -n $"Shutting down $prog: "
        killproc $ntpd
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/ntpd
        return $RETVAL
}

# See how we were called.
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  status)
        status $ntpd
        RETVAL=$?
        ;;
  restart)
        stop
        start
        RETVAL=$?
        ;;
  condrestart)
        if [ -f /var/lock/subsys/ntpd ]; then
            stop
            start
            RETVAL=$?
        fi
        ;;
  *)
        echo $"Usage: $0 {start|stop|status|restart|condrestart}"
        exit 1
esac
exit $RETVAL
```

Step 5
Finally, we must test the new chrooted jail configuration of our NTP server.

- Start the new chrooted jail NTP software with the following command:
```
[root@deep /]# /etc/init.d/ntpd start
Starting NTPD:               [OK]
```

- If you don't get any errors, do a '**ps ax | grep ntpd**' and see if we're running:
```
[root@deep /]# ps ax | grep ntpd
1349 ?        SL     0:00 /usr/sbin/ntpd -T /chroot/ntpd -l
/var/log/messages -
```

If so, let's check to make sure it's chrooted by picking out its process numbers and doing '**ls -la /proc/that_process_number/root/**'.

```
[root@deep /]# ls -la /proc/1349/root/
```

If you see something like:

```
total 4
drwxr-xr-x    4 root     root         1024 Jun 20 18:00 .
drwxr-xr-x    4 root     root         1024 Jun 20 17:12 ..
drwxr-xr-x    2 root     root         1024 Jun 20 17:15 etc
drwxr-xr-x    3 root     root         1024 Jun 20 18:00 var
```

Congratulations! Your NTP server in chroot jail is working.

NTP Administrative Tools

The commands listed below are some that we use often, but many more exist. Check the manual pages of NTP and documentation for more information.

ntpq

The command line utility ntpq can be used to check the status of a NTP daemon on either the local machine or on a remote host. It can be run in an interactive mode or in batch mode. In batch mode, it executes a command and returns to the command prompt.

- To print the status of a NTP daemon, use the following command:

```
[root@deep /]# ntpq -p

     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*ntp1.laas.fr    horlogegps.rese  2 u  222  256  377  121.853    4.313   0.540
 LOCAL(0)        LOCAL(0)        10 l   21   64  377    0.000    0.000   0.008
```

The table above shows the output for a NTP daemon which has 2 reference time sources: its own local clock, and a remote Time Server, with host address ntpl.laas.fr.

ntptrace

The ntptrace utility can be used to find the best Time Server to use depending of our physical location. Its primary use is to determine where a given **N**etwork **T**ime **P**rotocol (NTP) server gets its time from, and follows the chain of NTP servers back to their master time source.

One interesting use of this small utility is to calculate distance and response time from remote public Time Server on the Internet. This allows us to best choose which Time Server is more accurate for our network and internal Time Server. In this way we can with certitude get the best Time Server for our systems.

- To trace NTP server, use the following command:

```
[root@deep /]# ntptrace 128.182.58.100
mailer1.psc.edu: stratum 2, offset 0.014334, synch distance 0.05246
otc1.psu.edu: stratum 1, offset -0.024693, synch distance 0.03029, refid
'WWV'
```

The resulting output should be read from left to right. With the above command, we can get information about the host name, the host stratum, the time offset between that host, the host synchronization distance, and the reference clock ID. All times are given in seconds.

ntpdc

The ntpdc is one of the most important utility commands with NTP. It's used to query the ntpd daemon about its current state and to request changes in that state. The program may be run either in interactive mode or controlled using command line arguments.

Much operation could be done with this NTP utility, the best to get an idea of available options, is to run the command in interactive mode and use the help option to list all available features with the program.

- To run ntpdc in interactive mode, use the following command:

```
[root@deep /]# ntpdc
ntpdc> help
Commands available:
addpeer      addrefclock  addserver    addtrap      authinfo
broadcast    clkbug       clockstat    clrtrap      controlkey
ctlstats     debug        delay        delrestrict  disable
dmpeers      enable       exit         fudge        help
host         hostnames    iostats      kerninfo     keyid
keytype      listpeers    loopinfo     memstats     monlist
passwd       peers        preset       pstats       quit
readkeys     requestkey   reset        reslist      restrict
showpeer     sysinfo      sysstats     timeout      timerstats
traps        trustedkey   unconfig     unrestrict   untrustedkey
version
ntpdc> quit
```

CHAPTER 27

Quota

IN THIS CHAPTER

1. Build a kernel with `Quota` support enable
2. Compiling - Optimizing & Installing `Quota`
3. Modifying the `/etc/fstab` file
4. Creating the `aquota.user` and `aquota.group` files
5. Assigning `Quota` for Users and Groups
6. `Quota` Administrative Tools

Linux Quota

Abstract

Quota is a system administration tool for monitoring and limiting users' and/or groups' disk usage, per file system. Two features of disk storage with the Quota tool are available to set limits: the first is the number of inodes (number of files) a user or a group of users may possess and the second is the number of disk blocks (amount of space in kilobytes) that may be allocated to a user or a group of users. With Quota, users are forced by the system administrator to not consume an unlimited amount disk space on a system. This program is handled on per user and per file system basis and must be set separately for each file system.

It is useful for SMTP and FTP servers where limitations must be applied on the user's directory, but can be used for any other purposes. It is your to decide where and how to use it. Depending of the type of SMTP or FTP servers that you install, it may or may not be required. For example, if you install Exim as your mail server and ProFTPD as your FTP server, then you don't need to have and use Quota because that software come with their own quota support. Therefore, check if your applications support and provides quota before installing and using Quota.

I highly recommend to NOT using this software if you can because it's not so very well written and often contains many bugs. In most case, we can use the quota disk support that comes with the service that we want to install on our server. Now, every good service under Linux provides their own quota disk support that is preferable to use instead of the Quota tools explained here. It's your to decide whenever you really need this software and my recommendation is not to use this tool as much as possible.

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, at personal discretion).
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: Yes
Latest Quota version number is 3.06

The procedures given in this chapter are likely to work on all Linux platforms, but we have only tested it on OpenNA Linux and Red Hat Linux.

Packages

The following is based on information listed by Quota as of 2002/06/06. Please regularly check http://sourceforge.net/projects/linuxquota/ for the latest status. We chose to install the required component from a source file because it provides the facility to fine tune the installation.

Source code is available from:
Quota Homepage: http://sourceforge.net/projects/linuxquota/
You must be sure to download: quota-3.06.tar.gz

Prerequisites

Quota requires that the software below is already installed on your system to be able to compile successfully. If this is not the case, you must install it. Please make sure you have this program installed on your machine before you proceed with this chapter.

- ✓ e2fsprogs-devel is required to build Quota on your system.
- ✓ tcp_wrappers is required to build Quota on your system.
- ✓ gettext is required to build Quota on your system.

Build a kernel with Quota support enable

The first thing you need to do is ensure that your kernel has been built with Quota support enabled. In the 2.4 kernel version you need ensure that you have answered **Y** to the following questions:

```
*Filesystems
*
Quota support (CONFIG_QUOTA) [N/y/?] y
```

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all the files installed on the system in the eventuality of an update in the future. To solve the problem, it is a good idea to make a list of files on the system before you install Quota, and then one afterwards, and then compare them using the **diff** utility to find out what files were placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep root]# find /* > Quota1
  ```
- And the following one after you install the software:
  ```
  [root@deep root]# find /* > Quota2
  ```
- Then use the following command to get a list of what changed:
  ```
  [root@deep root]# diff Quota1 Quota2 > Quota-Installed
  ```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. In the example above, we use the /root directory of the system to store all generated list files.

Compiling - Optimizing & Installing Quota

Below are the steps that you must make to configure, compile and optimize the Quota software before installing it onto your system. First off, we install the program as the user 'root' so as to avoid permissioning problems.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- This can be done with the following commands:
  ```
  [root@deep /]# cp quota-version.tar.gz /var/tmp/
  [root@deep /]# cd /var/tmp/
  [root@deep tmp]# tar xzpf quota-version.tar.gz
  ```

Step 2
After that, move into the newly created Quota directory.

- To move into the newly created Quota directory, use the following command:
```
[root@deep tmp]# cd quota-tools/
```

Step 3
Now it is time configure, compile and optimize it for our system.

- To configure and optimize Quota use the following compilation lines:
```
CFLAGS="-O2 -march=i686 -funroll-loops"; export CFLAGS
./configure \
--prefix=/usr \
--sysconfdir=/etc \
--localstatedir=/var \
--mandir=/usr/share/man \
--enable-rpc=no \
--enable-rpcsetquota=no
```

Step 4
At this stage the program is ready to be built and installed. We build Quota with the 'make' command and produce a list of files on the system before we install the software, and one afterwards, then compare them using the **diff** utility to find out what files were placed where and finally install Quota.

```
[root@deep quota-tools]# make
[root@deep quota-tools]# cd
[root@deep root]# find /* > Quota1
[root@deep root]# cd /var/tmp/quota-tools/
[root@deep quota-tools]# make install
[root@deep quota-tools]# cd
[root@deep root]# find /* > Quota2
[root@deep root]# diff Quota1 Quota2 > Quota-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

Step 5
Once the compilation, optimization and installation of the software has completed, we can free up some disk space by deleting the program tar archives and the related source directory, since they are no longer needed.

- To delete Quota and its related source directory, use the following commands:
```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf quota-tools/
[root@deep tmp]# rm -f quota-version.tar.gz
[root@deep tmp]# rpm -e e2fsprogs-devel
[root@deep tmp]# rpm -e tcp_wrappers
```

The rm command as used above will remove all the source files we have used to compile and install Quota. It will also remove the Quota compressed archive from the /var/tmp directory.

Modifying the `/etc/fstab` file

The `/etc/fstab` file contains information about various file systems installed on your Linux server. `Quota` must be enabled in the `fstab` file before you can use it. Since `Quota` must be set for each file system separately, and because in the `fstab` file, each file system is described on a separate line, `Quota` must be set on each of the separate lines in the `fstab` for which you want to enable `Quota` support.

Step 1
With the program `Quota`, depending on your needs, etc, you can enable `Quota` for users, groups or both (users and groups). For all examples below, we'll use the `/home` directory and shows you the three possibilities.

Possibility 1:
To enable user `Quota` support on a file system, edit your `fstab` file and add the "`usrquota`" option to the fourth field after the word "`defaults`" or any other options you may have set for this specific file system.

- Edit your **`fstab`** file (`vi /etc/fstab`) and as an example change:

```
LABEL=/home      /home   ext3    defaults 1 2
```

To read:

```
LABEL=/home      /home   ext3    defaults,usrquota 1 2
```

Possibility 2:
To enable group `Quota` support on a file system, edit your `fstab` file and add the "`grpquota`" option to the fourth field after the word "`defaults`" or any other options you may have set for this specific file system.

- Edit your **`fstab`** file (`vi /etc/fstab`) and as an example change:

```
LABEL=/home      /home   ext3    defaults 1 2
```

To read:

```
LABEL=/home      /home   ext3    defaults,grpquota 1 2
```

Possibility 3:
To enable both users `Quota` and group `Quota` support on a file system, edit your `fstab` file and add the "`usrquota`" and "`grpquota`" options to the fourth field after the word "`defaults`" or any other options you may have set for this specific file system.

- Edit your **`fstab`** file (`vi /etc/fstab`) and as an example change:

```
LABEL=/home      /home   ext3    defaults 1 2
```

To read:

```
LABEL=/home      /home   ext3    defaults,usrquota,grpquota 1 2
```

Step 2
Once you have made the necessary adjustments to the /etc/fstab file, it is time to inform the system about the modification.

- This can be done with the following command:
```
[root@deep /]# mount /home -oremount
```

Each file system that has been modified must be remounted with the command shown above. In our example we have modified the /home file-system on the server.

Creating the aquota.user and aquota.group files

In order for Quotas to be established, the root directory of the file system on which you want to enable Quota feature must contain a file, owned by root, called "**aquota.user**" if you want to use and set user Quota, and/or "**aquota.group**" if you want to use and set group Quota, or both if you want users and group Quota.

We must create, in the directory in which we want to have Quota feature enabled, the required quotafiles, this must be made with the "quotacheck" utility. In our example, we will create under the /home directory the file for user and group restrictions as shown below.

- To create the aquota.user and/or aquota.group files, use the following commands:
```
[root@deep /]# touch /home/aquota.user
[root@deep /]# touch /home/aquota.group
[root@deep /]# chmod 0600 /home/aquota.user
[root@deep /]# chmod 0600 /home/aquota.group
[root@deep /]# quotacheck -u -a
[root@deep /]# quotacheck -g -a
```

The above commands will create the required quotafiles for us. In the first command, the "-u" option inform quota to create the file for users and the "-g" option will do it for group, finally the "-a" option means to do it for all file-system where quota feature is enable.

WARNING: Both Quota record files, aquota.user and aquota.group, should be owned by root, with read-write permission for "root" only (0600/-rw-------).

Assigning Quota for Users and Groups

After the required files have been created, you can assign Quotas to users or groups of users on your system. This operation is performed with the edquota tool.

Assigning quota for a particular user:

The edquota program is a Quota editor that creates a temporary file of the current disk Quotas to set Quotas for users or group of users in the system. The example below shows you how to setup Quotas for users or groups on your system.

Step 1
Consider, for example, that you have a user with the login id "admin" on your system. The following command takes you into the editor (vi) to edit and set Quotas for user "admin" on each partition that has Quotas enabled.

- To edit and modify Quota for user "admin", use the following command:

```
[root@deep /]# edquota -u admin
Disk quotas for user admin (uid 500):
  Filesystem     blocks     soft     hard     inodes     soft     hard
  /dev/sda7      16         0        0        4          0        0
```

After the execution of the above command, you will see the following lines related to the example user "admin" appear on the screen.

1. The "**blocks**" parameter display the total number of blocks (in kilobytes) the user has presently consumed on a partition.

2. The "**inodes**" value displays the total number of files the user has presently on a partition.

These parameters "blocks" and "inodes" are controlled and set automatically by the system and you don't need to touch them.

- To assign 5MB of quota for user "admin", change the following parameters:

```
Disk quotas for user admin (uid 500):
  Filesystem     blocks     soft     hard     inodes     soft     hard
  /dev/sda7      16         0        0        4          0        0
```

To read:

```
Disk quotas for user admin (uid 500):
  Filesystem     blocks     soft     hard     inodes     soft     hard
  /dev/sda7      16         5000     6000     4          0        0
```

1. The **soft** parameter specifies the maximum amount of disk usage a Quota user is allowed to have (in our example this amount is fixed to 5MB).

2. The **hard** parameter specifies the absolute limit on the disk usage a Quota user can't go beyond it.

The Grace period parameter:

The Grace period parameter allows you to set a time limit before the **soft** limit value is enforced on a file system with Quota enabled.

Step 1
For example, this parameter can be used to warn your users about a new policy that will set a Quota of 5MB of disk space in their home directory in 14 days. You can set the 7 days default part of this parameter to any length of time that you feel reasonable. The change of this setting requires two steps as follows (in my example I assume 14 days).

- Edit the default Grace period parameter, by using the following command:

```
[root@deep /]# edquota -t
Grace period before enforcing soft limits for users:
Time units may be: days, hours, minutes, or seconds
  Filesystem             Block grace period     Inode grace period
  /dev/sda7                     7days                  7days
```

- To modify the Grace period to 14 days. Change or set the following default parameters:

```
Grace period before enforcing soft limits for users:
Time units may be: days, hours, minutes, or seconds
  Filesystem             Block grace period     Inode grace period
  /dev/sda7                     7days                  7days
```

To read:

```
Grace period before enforcing soft limits for users:
Time units may be: days, hours, minutes, or seconds
  Filesystem             Block grace period     Inode grace period
  /dev/sda7                    14days                 14days
```

Assigning `quota` for a particular group:

Consider, for example, you have a group with the group id "`users`" on your system. The following command takes you into the `vi` editor to edit `Quotas` for the group "`users`" on each partition that has `Quotas` enabled:

- To edit and modify `Quota` for group "`users`", use the following command:

```
[root@deep /]# edquota -g users
Disk quotas for group users (gid 100):
  Filesystem     blocks     soft     hard     inodes     soft     hard
  /dev/sda7      16         0        0        4          0        0
```

The procedure is the same as for assigning `Quotas` for a particular user; as described previously, you must modify the parameter of "`soft`" and "`hard`" then save your change.

Assigning `quota` for groups of users with the same value:

The `edquota` tool has a special option `(-p)` that assign `Quotas` for groups of users with the same value assigned to an initial user. Assuming that you want to assign users starting at UID 500 on the system the same value as the user "`admin`", we would first edit and set admin's `Quota` information, then execute:

- To assign `Quota` for group of users with the same value, use the following command:

```
[root@deep /]# edquota -p admin `awk -F: '$3 > 499 {print $1}'
/etc/passwd`
```

The `edquota` program will duplicate the `Quota` that we have set for the user "`admin`" to all users in the `/etc/passwd` file that begin after UID 499.

NOTE : You can use the `quota` utility to set a maximun size to a mail box for your mail users. For example: set `quota` to users at 10M in your `/var` partition and put the min and max inodes parameter of quota to 1. Then a user will be able to keep in his `/var/mail/$LOGNAME` only 10M.

Quota Administrative Tools

The commands listed below are some that we use often, but many more exist. Check the manual page for more information.

quota

Quota displays users' disk usage and limits on a file system.

- To display user disk usage and limits, use the following command:

```
[root@deep /]# quota -u admin
Disk quotas for user admin (uid 500):
Filesystem  blocks  quota  limit  grace  files  quota  limit  grace
 /dev/sda7  16      5000   6000          4      0      0
```

- To display group Quotas for the group of which the user is member, use the command:

```
[root@deep /]# quota -g users
Disk quotas for group users (gid 100):
Filesystem  blocks  quota  limit  grace  files  quota  limit  grace
 /dev/sda7  16      5000   6000          4      0      0
```

repquota

The repquota utility produces summarized quota information of the disk usage and quotas for the specified file systems. Also, it prints the current number of files and amount of space used (in kilobytes) for each user.

- Here is a sample output repquota gives (you results may vary):

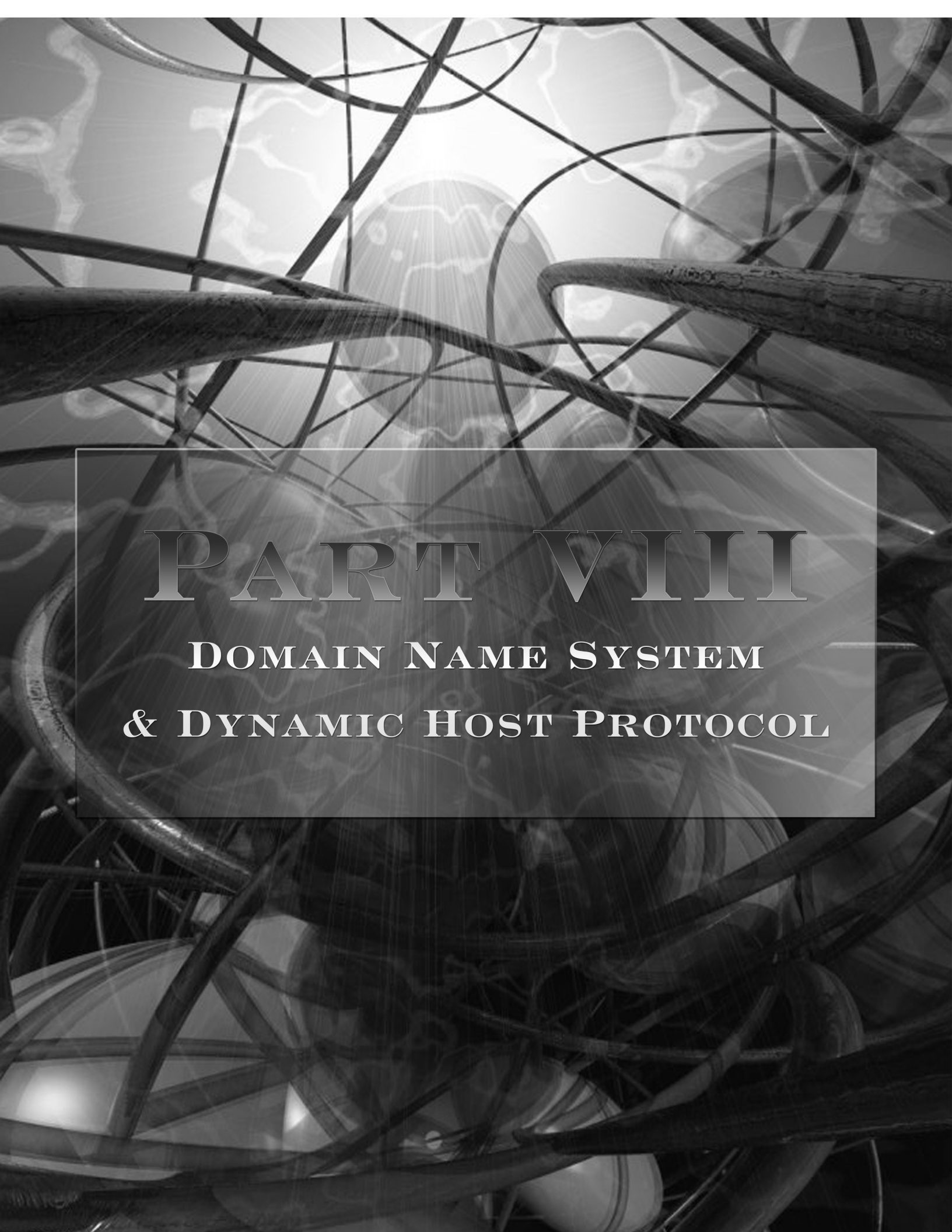
```
[root@deep /]# repquota -a

*** Report for user quotas on device /dev/sda7
Block grace time: 7days; Inode grace time: 7days
                        Block limits                File limits
User            used    soft    hard  grace    used  soft  hard  grace
----------------------------------------------------------------------
root      --   32828       0       0              5     0     0
admin     --      16    5000    6000              4     0     0
```

Further documentation

For more details, there are several manual pages about Quota that you could read:

```
$ man edquota (8)              - Edit user quotas.
$ man quota (1)                - Display disk usage and limits.
$ man quotacheck (8)           - Scan a file system for disk usages.
$ man quotactl (2)             - Manipulate disk quotas.
$ man quotaon, quotaoff (8)    - Turn file system quotas on and off.
$ man repquota (8)             - Summarize quotas for a file system.
$ man rquota (3)               - Implement quotas on remote machines.
```



Part VIII

Domain Name System & Dynamic Host Protocol

CHAPTER 28

ISC BIND & DNS

IN THIS CHAPTER

1. **Compiling - Optimizing & Installing `ISC BIND & DNS`**
2. **Configuring `ISC BIND & DNS`**
3. **Running `ISC BIND & DNS` as Caching-Only Name Server**
4. **Running `ISC BIND & DNS` as Primary Master Name Server**
5. **Running `ISC BIND & DNS` as Secondary Slave Name Server**
6. **Running `ISC BIND & DNS` in a chroot jail**
7. **Securing `ISC BIND & DNS`**
8. **Optimizing `ISC BIND & DNS`**
9. **`ISC BIND & DNS` Administrative Tools**
10. **`ISC BIND & DNS` Users Tools**

Linux ISC BIND & DNS

Abstract

Every time you send an electronic mail, surf the net, connect to another server, or talk with someone for example, you rely on the **D**omain **N**ame **S**ystem. It is rare that you don't have to pass through DNS in a networking environment. The **D**omain **N**ame **S**ystem is essential even if you don't run a **D**omain **N**ame **S**erver since it is the program (the directory to the Internet) that handles mapping between host names. Without it you cannot retrieve information remotely from anywhere on the network.

Domain **N**ame **S**ystem (DNS) is one of the **MOST** important network services for IP network communications, and for this reason, all Linux **client** machines should be configured to perform caching functions as a minimum. Setting up a caching server for client local machines will reduce the load on the site's primary server. A caching only name server will find the answer to name queries and remember the answer the next time we need it. This will significantly shorten the waiting time the next time the same query is made.

A **N**ame **S**erver (NS) is a program that stores information about named resources and responds to queries from programs called *resolvers*, which act as client processes. The basic function of an NS is to provide information about network objects by answering queries. Linux is a perfect platform to run and deploy the BIND DNS server; a number of Linux DNS servers on the Internet are listed as authoritative DNS servers for Microsoft's domains. Yes, Microsoft has partially outsourced the management of its **D**omain **N**ame **S**ystem (DNS) servers to Linux for the job. Oops.

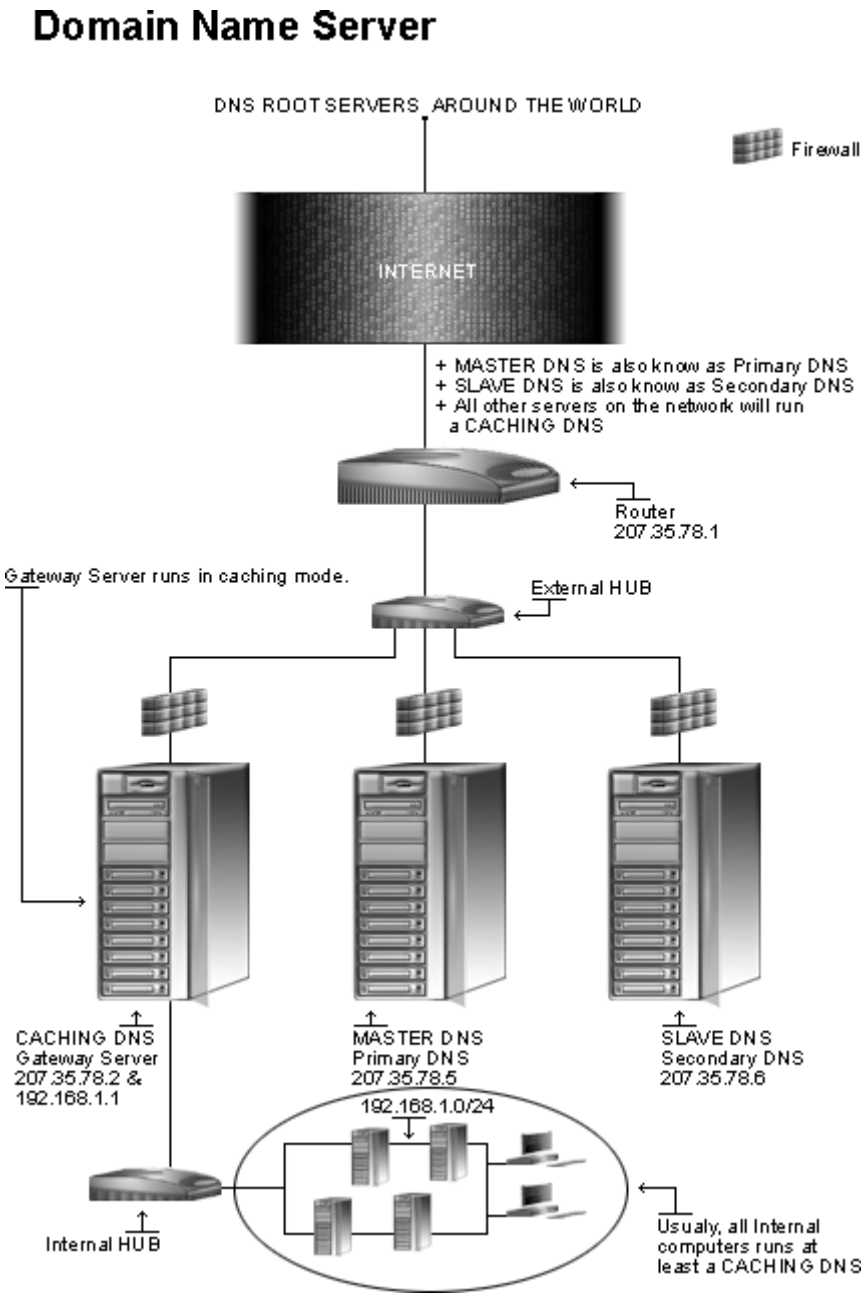
BIND (**B**erkeley **I**nternet **N**ame **D**omain) is a widely used, free implementation of the **D**omain **N**ame **S**ystem for Unix and Windows NT. It provides a server, a client library, and several utility programs. It is estimated to be the DNS software in use in over **90%** of the hosts on the Internet and this is the one that we will describe in this chapter.

To separate your internal **D**omain **N**ame **S**ervices from external DNS, it is better to use Split DNS, also known and referred to as "shadow namespaces". A Split DNS or "shadow namespace" is a name server that can answer queries from one source one way, and queries from another source another way. A Split DNS allow the Names, addresses and the topology of the secure network to be not available to the insecure external network. With Split DNS the external DNS only reveals public addresses and the internal DNS reveals internal IP addresses to the secure network. This is the recommended DNS configuration to use between hosts on the corporate network and external hosts.

To do split DNS, you must have two independent name servers for the same zone. One server and one copy of the zone are presented to the outside world. The other name server has a probably different bunch of contents for that zone which it makes available to the inside.

In our configuration and installation we'll run ISC BIND & DNS as non root-user and in a chrooted environment. We also provide you with three different configurations; one for a simple Caching Name Server Only (client), one for a Slave Name Server (Secondary DNS Server) and another one for a Master Name Server (Primary DNS Server).

The simple Caching Name Server configuration will be used for your servers that don't act as a Master or Slave Name Server, and the Slave and Master configurations will be used for your servers that act as a Master Name Server and Slave Name Server. Usually one of your servers acts as Primary/Master; another one acts as Secondary/Slave and the rest act as simple Caching client Name Servers.



This is a graphical representation of the DNS configuration we use in this book. We try to show you different settings (Caching Only DNS, Primary/Master DNS, and Secondary/Slave DNS) on different servers. Please note that lot of other possibilities exist, and depending on your needs, and network architecture design.

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, at personal discretion).
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest ICS BIND & DNS version number is 9.2.1

The procedures given in this chapter are likely to work on all Linux platforms, but we have only tested it on OpenNA Linux and Red Hat Linux.

Packages

The following is based on information listed by the ISC for BIND & DNS. Please regularly check http://www.isc.org/ for the latest status. We chose to install the required component from a source file because it provides the facility to fine tune the installation.

Source code is available from:
ISC BIND & DNS Homepage: http://www.isc.org/
ISC BIND & DNS FTP Site: 204.152.184.27
You must be sure to download: bind-9.2.1.tar.gz

Prerequisites

ICS BIND & DNS requires that the software below is already installed on your system to be able to compile successfully. If this is not the case, you must install it. Please make sure you have this program installed on your machine before you proceed with this chapter.

- ✓ OpenSSL is required to run ISC BIND & DNS with SSL support on your system.

NOTE: For more information on OpenSSL software, see its related chapter in this book.

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all the files installed on the system in the eventuality of an update in the future. To solve the problem, it is a good idea to make a list of files on the system before you install ISC BIND & DNS, and then one afterwards, and then compare them using the **diff** utility to find out what files were placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep root]# find /* > DNS1
  ```
- And the following one after you install the software:
  ```
  [root@deep root]# find /* > DNS2
  ```
- Then use the following command to get a list of what changed:
  ```
  [root@deep root]# diff DNS1 DNS2 > ISC-BIND-DNS-Installed
  ```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. In the example above, we use the /root directory of the system to store all generated list files.

Compiling - Optimizing & Installing ISC BIND & DNS

Below are the steps that you must make to configure, compile and optimize the ISC BIND & DNS software before installing it onto your system. First off, we install the program as the user 'root' so as to avoid permissioning problems.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- This can be done with the following commands:

```
[root@deep /]# cp bind-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf bind-version.tar.gz
```

Step 2
In order to check that the version of ISC BIND & DNS, which you are going to install, is an original and unmodified one, please check the supplied signature with the PGP key from ISC BIND & DNS. Unfortunately, ISC BIND & DNS doesn't provide a MD5 signature for verification. But a PGP key is available on the ISC BIND & DNS website.

To get a PGP key copy of ISC BIND & DNS, please point your browser to the following URL: http://www.isc.org/products/BIND/bind9.html. For more information about how to use this key for verification, see the GnuPG chapter in this book.

Step 3
ISC BIND & DNS needs a UID and GID to properly run on the system but this UID/GID cannot run as super-user root; for this reason we must create a special user with no shell privileges on the system for running ISC BIND & DNS daemon.

- To create this special BIND & DNS user on OpenNA Linux, use the following command:

```
[root@deep tmp]# groupadd -g 25 named > /dev/null 2>&1 || :
[root@deep tmp]# useradd -c "BIND DNS Server" -d /var/named -g 25 -s
/bin/false -u 25 named > /dev/null 2>&1 || :
```

- To create this special BIND & DNS user on Red Hat Linux, use the following command:

```
[root@deep tmp]# groupadd -g 25 named  2> /dev/null || :
[root@deep tmp]# useradd -u 25 -g 25 -s /bin/false -M -r -d /var/named
named > /dev/null 2>&1 || :
```

The above command will create a null account, with no password, no valid shell, no files owned-nothing but a UID and a GID for the program. Remember that ISC BIND & DNS daemon does not need to have a shell account on the server.

Step 4
Now, edit the **shells** file (`vi /etc/shells`) and add a non-existent shell name "`/bin/false`", which is the one we used in the `useradd` command above.

```
[root@deep tmp]# vi /etc/shells
/bin/bash2
/bin/bash
/bin/sh
/bin/false ← This is our added no-existent shell
```

Step 5
After that, move into the newly created `ISC BIND & DNS` directory and perform the following steps before compiling and optimizing it. The modifications we bring to the `ISC BIND & DNS` source file below are necessary to relocate some default files.

- To move into the newly created `ISC BIND & DNS` directory, use the following command:
 `[root@deep tmp]# cd bind-9.2.1/`

Step 6
The source file to modify is called **globals.h** and one of its functions is to specify the location of the `named.pid` and `lwresd.pid` files. We'll change the default location for these files to be compliant with our system.

- Edit the **globals.h** file (`vi +105 bin/named/include/named/globals.h`) and change the lines:

  ```
  "/run/named.pid");
  ```

 To read:

  ```
  "/run/named/named.pid");
  ```

 and

  ```
  "/run/lwresd.pid");
  ```

 To read:

  ```
  "/run/named/lwresd.pid");
  ```

Step 7
Once the required modifications have been made to the source file, it is time configure, compile and optimize it for our system.

- To configure and optimize `ISC BIND & DNS` use the following compilation lines:

```
CFLAGS="-O2 -march=i686 -funroll-loops"; export CFLAGS
./configure \
--prefix=/usr \
--sysconfdir=/etc \
--localstatedir=/var \
--mandir=/usr/share/man \
--with-openssl \
--with-libtool \
--disable-ipv6
```

Step 8
At this stage the program is ready to be built and installed. We build ISC BIND & DNS with the 'make' command and produce a list of files on the system before we install the software, and one afterwards, then compare them using the **diff** utility to find out what files were placed where and finally install ISC BIND & DNS.

```
[root@deep bind-9.2.1]# make
[root@deep bind-9.2.1]# cd
[root@deep root]# find /* > DNS1
[root@deep root]# cd /var/tmp/bind-9.2.1/
[root@deep bind-9.2.1]# make install
[root@deep bind-9.2.1]# strip /usr/sbin/named
[root@deep bind-9.2.1]# mkdir -p /var/named
[root@deep bind-9.2.1]# mkdir -p /var/run/named
[root@deep bind-9.2.1]# install -c -m0600 bin/rndc/rndc.conf /etc/
[root@deep bind-9.2.1]# chown named.named /etc/rndc.conf
[root@deep bind-9.2.1]# chown named.named /var/named/
[root@deep bind-9.2.1]# chown named.named /var/run/named/
[root@deep bind-9.2.1]# /sbin/ldconfig
[root@deep bind-9.2.1]# cd
[root@deep root]# find /* > DNS2
[root@deep root]# diff DNS1 DNS2 > ISC-BIND-DNS-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

Step 9
Once the compilation, optimization and installation of the software has completed, we can free up some disk space by deleting the program tar archive and the related source directory, since they are no longer needed.

- To delete ISC BIND & DNS and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf bind-version/
[root@deep tmp]# rm -f bind-version.tar.gz
```

The rm command as used above will remove all the source files we have used to compile and install ISC BIND & DNS. It will also remove the ISC BIND & DNS compressed archive from the /var/tmp directory.

Configuring ISC BIND & DNS

After ISC BIND & DNS has been built and installed successfully on your system, the next step is to configure and customize its configuration files to fit your needs.

- ✓ /etc/named.conf: (The ISC BIND & DNS Configuration File)
- ✓ /var/named/db.cache: (The Root Server Hints File)
- ✓ /var/named/db.localhost: (The Mapping File)
- ✓ /var/named/0.0.127.in-addr.arpa: (The Reverse Mapping File)
- ✓ /etc/sysconfig/named: (The ISC BIND & DNS System Configuration File)
- ✓ /etc/init.d/named: (The ISC BIND & DNS Initialization File)
- ✓ /var/named/db.openna: (Addr to Host Names Mapping File)
- ✓ /var/named/78.35.207.in-addr.arpa: (Host Names to Addr Mapping)

Running ISC BIND & DNS as Caching-Only Name Server

This section applies only if you chose to install and use ISC BIND & DNS as a Caching Name Server in your system. Caching-only name servers are servers not authoritative for any domains except 0.0.127.in-addr.arpa (the localhost). A Caching-Only Name Server can look up names inside and outside your zone, as can Primary and Slave Name Servers. The difference is that when a Caching-Only Name Server initially looks up a name within your zone, it ends up asking one of the Primary or Slave Names Servers for your zone for the answer.

Remember that a Caching-Only Name Server should be run on any systems which are not a Primary or Secondary Name Servers. This is why I begin the configuration of ISC BIND & DNS with the Caching-Only Name Server configuration. ISC BIND & DNS is very important and must be installed in all types of server, since many of the services described in this book rely on it to work properly. Without DNS servers no one on the Internet will be able to find your servers.

/etc/named.conf: The ISC BIND & DNS Configuration File

The /etc/named.conf file is the main configuration file for ISC BIND & DNS. It is in this configuration file that ISC BIND & DNS gets all of its network information. Use this configuration file for all servers on your network that don't act as a Master or Slave Name Server. Setting up a simple Caching Server for local client machines will reduce the load on the network's primary server.

Step 1
With this configuration for a Caching-Only Name Server, all queries from outside clients are refused. The text in bold are the parts of the configuration file that must be customized and adjusted to satisfy our needs.

- Create the **named.conf** file (touch /etc/named.conf) and add the following lines in the file. Below is what I recommend you set.

```
// Authorized source addresses.
acl "trusted" {
        localhost;
};

// Known fake source addresses shouldn't be replied to.
acl "bogon" {
        0.0.0.0/8;
        1.0.0.0/8;
        2.0.0.0/8;
        192.0.2.0/24;
        224.0.0.0/3;
        169.254.0.0/16;
        // Enterprise networks may or may not be bogus.
        10.0.0.0/8;
        172.16.0.0/12;
        192.168.0.0/16;
};

options {
        directory "/var/named";
        allow-transfer { none; };
        allow-query { trusted; };
        allow-recursion { trusted; };
        blackhole { bogon; };
        tcp-clients 32;
        forwarders { 207.35.78.5; 207.35.78.6; };
        version "OpenNA Linux";
};
```

```
logging {
        category lame-servers { null; };
};

// Root server hints
zone "." { type hint; file "db.cache"; };

// Provide a reverse mapping for the loopback address 127.0.0.1/24
zone "localhost" {
        type master;
        file "db.localhost";
        notify no;
};

zone "0.0.127.in-addr.arpa" {
        type master;
        file "0.0.127.in-addr.arpa";
        notify no;
};
```

This tells the `named.conf` file to set itself up for this particular configuration with:

```
acl "trusted" {
        localhost;
};
```

The `acl` statement sets up `ACL` to be used by `ISC BIND & DNS` and can appear more than once in a configuration file. We use this statement to define new **A**ccess **C**ontrol **L**ist that we want to apply under some part of our configuration file. This is useful to avoid replication of same values along the configuration of the `named.conf` file.

In the above `ACL` line, we define a new `ACL` called "`trusted`", which will handle all allowed `IP` addresses or host names for our configuration. In this way we can refer to it with just its name "`trusted`" and the software will automatically apply what we've defined inside this `ACL` name to the configuration. The value "`localhost`" is the only value we use inside this `ACL` definition. For a Caching-Only Name Server, this is enough to make it work.

```
acl "bogon" {
        0.0.0.0/8;
        1.0.0.0/8;
        2.0.0.0/8;
        192.0.2.0/24;
        224.0.0.0/3;
        169.254.0.0/16;
        // Enterprise networks may or may not be bogus
        10.0.0.0/8;
        172.16.0.0/12;
        192.168.0.0/16;
};
```

Here we define another `ACL` statement, but this time for all `IP` addresses that we want to deny access to our `DNS` server. As you can see I've called this `ACL` "`bogon`" and have added to it all `IP` addresses that should be denied access to the `DNS` server. The above `IP's` are what the `RFC` recommends us to block, but feel free to add any `IP` address that you want to block. We will refer to this `ACL` later in the configuration when we need it. This is a security feature.

```
Options {};
```
The `options` statement sets up global options to be used by `ISC BIND & DNS` and may appear only once in a configuration file.

```
directory "/var/named";
```
The `directory` statement indicates the working directory of the server and should be an absolute path. The working directory is where all configuration files related to `ISC BIND & DNS` reside.

```
allow-transfer { none; };
```
The `allow-transfer` statement specifies which hosts are allowed to receive zone transfers from the Primary/Master Name Server. The default setting of `ISC BIND & DNS` is to allow transfers from all hosts. Since zone transfer requests are only required for Secondary/Slave Name Servers and since the configuration we are trying to do here is for a Caching-Only Name Server, we can completely disable this directive with the value "`none`". This is a security feature.

```
allow-query { trusted; };
```
The `allow-query` statement specifies which hosts are allowed to ask ordinary questions to the Caching Name Server. The default setting in the `options` block is to allow queries from all hosts. In configuring a Caching-Only Name Server, we should allow queries from the localhost only. Note that I use the `ACL` called "`trusted`" here to define the list of allowed hosts. This is a security feature.

```
allow-recursion { trusted; };
```
The `allow-recursion` statement specifies which hosts are allowed to make recursive queries through this server. With the configuration shown above, we allow recursive queries only from localhost since allowing external hosts on the Internet to ask your name server to answer recursive queries can open you up to certain kinds of cache poisoning attacks. Again, I use here the `ACL` called "`trusted`" to define the list of allowed hosts. This is a security feature.

```
blackhole { bogon; };
```
The `blackhole` statement specifies which hosts are NOT allowed to make any kind of queries through this server. With the configuration as shown above, we block all `IP's` listed inside the "`bogon`" `ACL`. This is a security feature.

```
tcp-clients 32;
```
The `tcp-clients` statement is used to define the maximum number of simultaneous client `TCP` connections the `DNS` server will accept, this is useful to control server resource limits and to avoid some kind of `DoS` attacks. On a Caching-Only Name server, we can set the value to a low number and on Primary or Secondary Name Servers we should set this value to something like 1024 to improve performance. This is a security and optimization feature.

```
forwarders { 207.35.78.5; 207.35.78.6; };
```
The `forwarders` statement specifies the `IP` addresses to be used for forwarding. Servers that do not have direct access to the Internet can use this option to create a large site-wide cache, reducing traffic over links to external name servers and to allow queries. It occurs only on those queries for which the server is not authoritative and does not have the answer in its cache. In the "`forwarders`" line, `207.35.78.5` and `207.35.78.6` are the `IP` addresses of our Primary (Master) and Secondary (Slave) `DNS` servers. They can also be the `IP` addresses of your ISP's `DNS` server and another `DNS` server, respectively.

Why would one assume that what's in one's ISP's name server's cache is any more "secure" than what one gets from the authoritative servers directly? That makes no sense at all. ISP's are often lazy about upgrades, which mean that there's a substantial risk that their name servers may be compromised or cache-poisoned. Another downside of forwarding, of course, is that it introduces an extra hop for *every* query which can't be satisfied from the local server's cache or authoritative data.

Now, sometimes that hop is worth it (because the answer is in your forwarder's cache, so you don't need to expend other "hops" over the Internet trying to resolve it yourself), but at other times (when the answer *doesn't* happen to be in the forwarders cache), it just adds latency. So forwarding can *sometimes* be justified in terms of query performance. But in this case, it should be configured as "forward first" to provide redundancy in case the forwarders are unavailable. This is the default value "forward first" into `BIND9`, and causes the server to query the `IP` addresses as specified in the forwarders statement (the forwarders first), and if that doesn't answer the question, the server will then look for the answer itself. This is a performance feature.

```
version "OpenNA Linux";
```

The `version` statement allows us to hide the real version number of our `ISC BIND & DNS` server. This can be useful when someone from the Internet tries to scan our **D**omain **N**ame **S**erver for possible vulnerable versions of the software. You can change the string "`OpenNA Linux`" to whatever you want. Note doing this will not prevent attacks and may impede people trying to diagnose problems with your server. This is a security feature.

```
logging {
        category lame-servers { null; };
};
```

The `logging` statement allows us to configure logging so that `lame-server` message aren't logged, which will reduce the overhead on your `DNS` and `syslog` servers. Lame-server messages are hosts that are believed to be name servers for the given domains, but which do not believe them selves to be such. This is often due to a configuration error on the part of that host master. It is a good idea to use the above option to completely disable this king of bogus message from our log file.

```
zone "." { type hint; file "db.cache"; };
```

All statements and definitions we have used in the configuration file so far were related to the way we wanted to customize and configure the software. Nothing has been added to inform `BIND` about how we want it to run on this system.

The "`zone`" statement is made for this purpose and depending of the type of `DNS` server that you want to run, its definitions and parameters will be significantly different. For a Caching-Only Name Server, the implementation is really not difficult to setup but for a Primary or Secondary Name Server, the file can become very large.

In all cases, the "`zone`" statement refers to another file on our system where all information related to zones on our network is provided. Also, the "`zone`" statement can have specific options that will inform the software about how we want it to manage the zone in question.

In every configuration of `ISC BIND & DNS`, we should have at least a "`zone`" statement definition for the localhost of the system on which it is running as well as definition of the root `DNS` server on the Internet. Here we provide this definition for the root `DNS` server by referring the software to get the information from the file called "`db.cache`" available under the `/var/named` directory.

```
zone "localhost" {
        type master;
        file "db.localhost";
        notify no;
};
```

The above "zone" statement is used to provide a mapping for the localhost address on the system. The "type master" option informs the software that the "db.localhost" file is the master file for this zone and the "notify no" option is used to avoid transfers of this localhost configuration file to other Name Servers. This is a security feature.

```
zone "0.0.127.in-addr.arpa" {
        type master;
        file "0.0.127.in-addr.arpa";
        notify no;
};
```

The above "zone" statement completes our configuration for a Caching-Only Name Server and it is used to provide a reverse mapping for the loopback address 127.0.0.1/24 on the system. As for the above definition, the "type master" option informs the software that the "0.0.127.in-addr.arpa" file is the master file for this zone and the "notify no" option is used to avoid transfer of this localhost configuration file to other Name Servers. This is a security feature.

Step 2
Now, set the permissions of the named.conf file to be (0600/-rw-------) and owned by the user 'named' with group permission set to "named" user for security reason.

- To change the permission mode and ownership of the named.conf file, use:
  ```
  [root@deep /]# chmod 600 /etc/named.conf
  [root@deep /]# chown named.named /etc/named.conf
  ```

/var/named/db.cache: The Root Server Hints File

This section applies to all types of Name Server (Caching, Master or Slave) that you may want to install on your system. The **db.cache** file is also know as the "Root Server Hints File" and tells your server (Caching, Master or Slave) where the servers for the root zone "." are, you must get a copy of the **db.cache** file and copy this file into the /var/named directory. All types of DNS server need this configuration file to work properly.

Step 1
Use the following commands on another Unix computer in your organization to query a new **db.cache** file for your Name Servers or pick one from your Linux CD-ROM source distribution:

- To query a new db.cache file, use the following command:
  ```
  [root@deep named]# dig @a.root-servers.net . ns > db.cache
  ```

- To query a new db.cache file by IP address, use the following command:
  ```
  [root@deep named]# dig @198.41.0.4 . ns > db.cache
  ```

A db.cache file should look like the following. If you want, you can use this one to start.

```
; <<>> DiG 9.2.1 <<>> @198.41.0.4 . ns
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 55980
;; flags: qr aa rd; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 13

;; QUESTION SECTION:
;.                              IN      NS

;; ANSWER SECTION:
.                       518400  IN      NS      I.ROOT-SERVERS.NET.
.                       518400  IN      NS      E.ROOT-SERVERS.NET.
.                       518400  IN      NS      D.ROOT-SERVERS.NET.
.                       518400  IN      NS      A.ROOT-SERVERS.NET.
.                       518400  IN      NS      H.ROOT-SERVERS.NET.
.                       518400  IN      NS      C.ROOT-SERVERS.NET.
.                       518400  IN      NS      G.ROOT-SERVERS.NET.
.                       518400  IN      NS      F.ROOT-SERVERS.NET.
.                       518400  IN      NS      B.ROOT-SERVERS.NET.
.                       518400  IN      NS      J.ROOT-SERVERS.NET.
.                       518400  IN      NS      K.ROOT-SERVERS.NET.
.                       518400  IN      NS      L.ROOT-SERVERS.NET.
.                       518400  IN      NS      M.ROOT-SERVERS.NET.

;; ADDITIONAL SECTION:
I.ROOT-SERVERS.NET.     3600000 IN      A       192.36.148.17
E.ROOT-SERVERS.NET.     3600000 IN      A       192.203.230.10
D.ROOT-SERVERS.NET.     3600000 IN      A       128.8.10.90
A.ROOT-SERVERS.NET.     3600000 IN      A       198.41.0.4
H.ROOT-SERVERS.NET.     3600000 IN      A       128.63.2.53
C.ROOT-SERVERS.NET.     3600000 IN      A       192.33.4.12
G.ROOT-SERVERS.NET.     3600000 IN      A       192.112.36.4
F.ROOT-SERVERS.NET.     3600000 IN      A       192.5.5.241
B.ROOT-SERVERS.NET.     3600000 IN      A       128.9.0.107
J.ROOT-SERVERS.NET.     3600000 IN      A       198.41.0.10
K.ROOT-SERVERS.NET.     3600000 IN      A       193.0.14.129
L.ROOT-SERVERS.NET.     3600000 IN      A       198.32.64.12
M.ROOT-SERVERS.NET.     3600000 IN      A       202.12.27.33

;; Query time: 25 msec
;; SERVER: 198.41.0.4#53(198.41.0.4)
;; WHEN: Tue May 28 11:07:43 2002
;; MSG SIZE  rcvd: 436
```

NOTE: Don't forget to copy the `db.cache` file to the `/var/named` directory on your Name Server after retrieving it from the Internet. The root name servers on the Internet do not change very often, but they do change occasionally. A good practice is to update your `db.cache` file every month or two.

Step 2
Now, set the permissions of the `db.cache` file to be (`0644/-rw-r--r--`) and owned by the user '`named`' with group permission set to "`named`" user for security reason.

- To change the permission mode and ownership of the `db.cache` file, use:
  ```
  [root@deep /]# chmod 644 /var/named/db.cache
  [root@deep /]# chown named.named /var/named/db.cache
  ```

/var/named/db.localhost: The Mapping File

This section applies to all types of Name Server (Caching, Master or Slave) that you may want to install on your system. The "db.localhost" file covers the localhost network on your system by providing a mapping for the localhost address on your server. All types of DNS server need this configuration file to properly work.

Step 1
Create the file in question under your /var/named directory.

- Create the **db.localhost** file (touch /var/named/db.localhost) and add the following lines inside the file:

```
$TTL 86400
@               IN      SOA     localhost. root.localhost. (
                                00      ; Serial
                                10800   ; Refresh after 3 hours
                                3600    ; Retry after 1 hour
                                604800  ; Expire after 1 week
                                86400 ) ; Minimum

                IN      NS      localhost.

localhost       IN      A       127.0.0.1
```

Step 2
Now, set the permissions of the db.localhost file to be (0644/-rw-r--r--) and owned by the user 'named' with group permissions set to "named" user for security reasons.

- To change the permission mode and ownership of db.localhost file, use:

```
[root@deep /]# chmod 644 /var/named/db.localhost
[root@deep /]# chown named.named /var/named/db.localhost
```

/var/named/0.0.127.in-addr.arpa: The Reverse Mapping File

This section applies to all types of Name Server (Caching, Master or Slave) that you may want to install on your system. The "0.0.127.in-addr.arpa" file covers the loopback network by providing a reverse mapping for the loopback address on your system. All types of DNS server need this configuration file to properly work.

Step 1
Create the file in question under your /var/named directory.

- Create the **0.0.127.in-addr.arpa** file (touch /var/named/0.0.127.in-addr.arpa) and add the following lines in the file:

```
$TTL 86400
@       IN      SOA     localhost. root.localhost. (
                                00      ; Serial
                                10800   ; Refresh after 3 hours
                                3600    ; Retry after 1 hour
                                604800  ; Expire after 1 week
                                86400 ) ; Minimum

        IN      NS      localhost.
1       IN      PTR     localhost.
```

Step 2
Now, set the permissions of the `0.0.127.in-addr.arpa` file to be (`0644/-rw-r--r--`) and owned by the user '`named`' with group permissions set to "`named`" user for security reasons.

- To change the permission mode and ownership of the `0.0.127.in-addr.arpa` file, use:

```
[root@deep /]# chmod 644 /var/named/0.0.127.in-addr.arpa
[root@deep /]# chown named.named /var/named/0.0.127.in-addr.arpa
```

/etc/sysconfig/named: The ISC BIND & DNS System Configuration File

This section applies to all types of Name Server (Caching, Master or Slave) that you may want to install on your system. The `/etc/sysconfig/named` file is used to specify `ISC BIND & DNS` system configuration information, such as if `ISC BIND & DNS` should run in a chroot environment, and if additional options are required to be passed to `named` daemon at startup.

- Create the **named** file (`touch /etc/sysconfig/named`) and add the following lines:

```
# This option will run named in a chroot environment.
#ROOTDIR="/chroot/named/"

# These additional options will be passed to named at startup.
# Don't add -t here, use ROOTDIR instead.
#OPTIONS=""
```

The "`ROOTDIR`" option instructs `ISC BIND & DNS` where its root directory should be located; this line is useful when you want to run `ISC BIND & DNS` in a chroot jail environment for increased security. For now, this line must be commented out since we'll see later in this chapter how to run `ISC BIND & DNS` in a chroot environment and how to use this option.

The "`OPTIONS`" parameter can be used to add the "`-d`" option for debug level of `ISC BIND & DNS` but in most cases we don't need to use it.

/etc/init.d/named: The ISC BIND & DNS Initialization File

This section applies to all types of Name Server (Caching, Master or Slave) that you may want to install on your system. The `/etc/init.d/named` script file is responsible for automatically starting and stopping the `DNS` server. Loading the `named` daemon as a standalone daemon will eliminate load time and will even reduce swapping since non-library code will be shared.

Please note that the following script is only suitable for Linux OS that use `System V`. If your Linux system uses some other method, like `BSD`, you'll have to adjust the script below to make it work for you.

Step 1
Create the **named** script file (`touch /etc/init.d/named`) and add the following lines:

```
#!/bin/bash

# This shell script takes care of starting and stopping named.
#
# chkconfig: 2345 55 45
# description: Named (BIND) is a Domain Name Server (DNS) that is used \
#              to resolve host names to IP addresses.
#
# processname: named
```

```
# Source function library.
. /etc/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Source for additional options if we have them.
if [ -f /etc/sysconfig/named ] ; then
        . /etc/sysconfig/named
fi

# Check that networking is up.
[ "${NETWORKING}" = "no" ] && exit 0

# If Named is not available stop now.
[ -f /usr/sbin/named ] || exit 0
[ -f "${ROOTDIR}"/etc/named.conf ] || exit 0

# Path to the Named binary.
named=/usr/sbin/named

RETVAL=0
prog="Named"

start() {
        echo -n $"Starting $prog: "
        if [ -n "${ROOTDIR}" -a "x${ROOTDIR}" != "x/" ]; then
                OPTIONS="${OPTIONS} -t ${ROOTDIR}"
        fi
        daemon $named -u named ${OPTIONS}
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/named
        return $RETVAL
}

stop() {
        echo -n $"Shutting down $prog: "
        killproc $named
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/named
        return $RETVAL
}

# See how we were called.
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  status)
        status $named
        RETVAL=$?
        ;;
  restart)
        stop
        start
        RETVAL=$?
        ;;
```

```
  condrestart)
        if [ -f /var/lock/subsys/named ]; then
                stop
                start
                RETVAL=$?
        fi
        ;;
  reload)
        /usr/sbin/rndc reload >/dev/null 2>&1 || /usr/bin/killall -HUP $named
        return $RETVAL
        ;;
  probe)
        /usr/sbin/rndc reload >/dev/null 2>&1 || echo start
        return $RETVAL
        ;;
  *)
        echo $"Usage: $0 {start|stop|status|restart|condrestart|reload|probe}"
        exit 1
esac
exit $RETVAL
```

Step 2
Once the `/etc/init.d/named` script file has been created, it is important to make it executable, change its default permissions, create the necessary links and then start it. Making this file executable will allow the system to run it, changing its default permissions to allow only the root user to change it for security reasons, and the creation of the symbolic links will let the process control initialization of Linux, which is in charge of starting all the normal and authorized processes that need to run at boot time on your system, start the program automatically for you at each system reboot.

- To make this script executable and to change its default permissions, use the commands:
  ```
  [root@deep /]# chmod 700 /etc/init.d/named
  [root@deep /]# chown 0.0 /etc/init.d/named
  ```

- To create the symbolic `rc.d` links for `ISC BIND & DNS`, use the following commands:
  ```
  [root@deep /]# chkconfig --add named
  [root@deep /]# chkconfig --level 2345 named on
  ```

- To start `ISC BIND & DNS` software manually, use the following command:
  ```
  [root@deep /]# /etc/init.d/named start
  Starting Named:          [OK]
  ```

Running `ISC BIND & DNS` as Primary Master Name Server

This section applies only if you chose to install and use `ISC BIND & DNS` as a Primary Name Server on your system. The Primary Master Name Server is the ultimate source of information about a domain. The Primary Master is an authoritative server configured to be the source of zone transfers for one or more Secondary Name Servers. The Primary Master Name Server obtains data for the zone from a file on disk.

`/etc/named.conf`: The `ISC BIND & DNS` Configuration File

Use this `named.conf` configuration file for the server on your network that acts as a Master Name Server. In every respectable networking environment, you need to set up at least a Primary Domain Name Server for your network. We'll use "`openna.com`" as an example domain, and assume you are using `IP` network address of `207.35.78.0`. Of course, you have to change all of these `IP` network address and example domains to fit your own parameters.

Step 1
To do this, add/change the following lines to your `/etc/named.conf` file. Text in bold are the parts of the configuration file that change from the previous `named.conf` file. Don't forget to adjust the values to satisfy your needs.

- Create the **`named.conf`** file (`touch /etc/named.conf`) and add the following lines in the file. Below is what I recommend you set.

```
// Authorized source addresses.
acl "trusted" {
        localhost;
        192.168.1.0/24;
        207.35.78.0/24;
};

// Known fake source addresses shouldn't be replied to.
acl "bogon" {
        0.0.0.0/8;
        1.0.0.0/8;
        2.0.0.0/8;
        192.0.2.0/24;
        224.0.0.0/3;
        169.254.0.0/16;
        // Enterprise networks may or may not be bogus.
        10.0.0.0/8;
        172.16.0.0/12;
        192.168.0.0/16;
};

options {
        directory "/var/named";
        allow-transfer { 207.35.78.6; };
        allow-query { trusted; };
        allow-recursion { trusted; };
        blackhole { bogon; };
        tcp-clients 1024;
        forwarders { none; };
        version "OpenNA Linux";
};

logging {
        category lame-servers { null; };
};

// Root server hints
zone "." { type hint; file "db.cache"; };

// Provide a reverse mapping for the loopback address 127.0.0.1/24
zone "localhost" {
        type master;
        file "db.localhost";
        notify no;
};

zone "0.0.127.in-addr.arpa" {
        type master;
        file "0.0.127.in-addr.arpa";
        notify no;
};

// We are the master server for OpenNA.com
zone "openna.com" {
```

```
        type master;
        file "db.openna";
        allow-query { any; };
};

// Provide a reverse mapping for domains network 207.35.78.0/27
zone "78.35.207.in-addr.arpa" {
        type master;
        file "78.35.207.in-addr.arpa";
        allow-query { any; };
};
```

This tells the `named.conf` file to set itself up for this particular configuration with:

```
acl "trusted" {
        localhost;
        192.168.1.0/24;
        207.35.78.0/24;
};
```

Here we change the default ACL statement called "trusted" to add the IP addresses of our internal private network we want to allow to use the Primary DNS Server for all kinds of name resolution. You should only list inside this ACL statement, allowed hosts. This is a security feature.

```
allow-transfer { 207.35.78.6; };
```

The allow-transfer statement specifies which hosts are allowed to receive zone transfers from the Primary/Master Name Server. The default setting of ISC BIND & DNS is to allow transfers to all hosts. In the allow-transfer line as shown above, 207.35.78.6 (our Secondary/Slave Name Server) is the only IP address allowed to receive zone transfers from the Primary/Master Name Server. You should configure your server to respond to zone transfers requests only from authorized IP addresses. In most cases, you'll only authorize your known Slave Servers to transfer zones from your Primary/Master Name Server. As the information provided is often used by spammers and IP spoofers. This is a security feature.

```
tcp-clients 1024;
```

The tcp-clients statement is used to define the maximum number of simultaneous client TCP connections the DNS server will accept, this is useful to control server resource limits. On a Primary Name server, we should set the value to a high number to improve performance. This is a security and optimization feature.

```
forwarders { none; };
```

The forwarders statement specifies the IP addresses to be used for forwarding. Servers that do not have direct access to the Internet can use this option to create a large site-wide cache, reducing traffic over links to external name servers and to allow queries. It occurs only on those queries for which the server is not authoritative and does not have the answer in its cache.

Since we are configuring BIND to run as a Primary Name Server in this configuration file, it is not required at all to define and use a "forwarders" statement here because a Primary Name Server is the ultimate source for domain name information and it doesn't need to forward queries to other servers to know about your domain name. Its sole purpose is to know about your domain name. We disable this option with "none" inside the statement.

```
version "OpenNA Linux";
```

The `version` statement allows us to hide the real version number of our `ISC BIND & DNS` server. This can be useful when someone from the Internet tries to scan our **D**omain **N**ame **S**erver for possible vulnerable versions of the software. You can change the string "`OpenNA Linux`" to whatever you want. Note: doing this will not prevent attacks and may impede people trying to diagnose problems with your server. This is a security feature.

```
zone "openna.com" {
        type master;
        file "db.openna";
        allow-query { any; };
};
```

The above "`zone`" statement is what makes our Primary Name Server the Master Name Server for our domain name "`openna.com`". As usual, the "`zone`" definition informs `BIND` that the domain name "`openna.com`" is under its jurisdiction, the "`type`" definition means this `DNS` server is the master server for "`openna.com`", the "`file`" definition informs the software where it can find the file that handles of all the domain information and finally, the "`alow-query`" means that every external host can ask our Primary Name Server about information on the domain name called "`openna.com`".

This is the way we define domain names we hope to add to our Primary Domain Name Server. If I have another domain name to add to my Primary Name Server, I will do it the same way as shown above, but will certainly change the name of the domain and db file to reflect the one associated with the new domain.

```
zone "78.35.207.in-addr.arpa" {
        type master;
        file "78.35.207.in-addr.arpa";
        allow-query { any; };
};
```

The above "`zone`" statement completes our configuration for a Primary Domain Name Server and it is used to provide a reverse mapping for all domains addresses 207.35.78.0/27 on the system. As for the above definition, the "`type master`" option informs the software that "`78.35.207.in-addr.arpa`" file is the master file for this zone and the "`alow-query`" means that all external hosts can ask our Primary Name Server about information on `IP` addresses ranges "207.35.78.0/27".

Step 2

Now, set the permissions of the `named.conf` file to be (`0600/-rw-------`) and owned by the user '`named`' with group permissions set to "`named`" user for security reasons.

- To change the permission mode and ownership of the `named.conf` file, use:

```
[root@deep /]# chmod 600 /etc/named.conf
[root@deep /]# chown named.named /etc/named.conf
```

/var/named/db.openna: Addr to Host Names Mapping File

This section applies only if you chose to install and use ISC BIND & DNS as a Primary Name Server in your system. Use this configuration file for the server on your network that acts as a Master Name Server. The "db.openna" file maps addresses to host names.

Step 1
Create the file in question under your /var/named directory.

- Create the **db.openna** file (touch /var/named/db.openna) and add the following lines in the file:

```
;$ORIGIN openna.com.
$TTL 172800
@  IN  SOA  ns1.openna.com. root.openna.com.  (
                                        01      ; Serial
                                        10800   ; Refresh after 3 hours
                                        3600    ; Retry after 1 hour
                                        604800  ; Expire after 1 week
                                        172800 ); Minimum TTL of 2 days

; Name Servers (NS) records.
;
        IN      NS      ns1.openna.com.
        IN      NS      ns2.openna.com.

; Mail Exchange (MX) records.
;
        MX      0       smtp.openna.com.

; Addresses for the canonical names (A) records.
;
localhost       IN      A       127.0.0.1
router          IN      A       207.35.78.1
gtw             IN      A       207.35.78.2
www             IN      A       207.35.78.3
smtp            IN      A       207.35.78.4
```

Step 2
Now, set the permissions of the db.openna file to be (0644/-rw-r--r--) and owned by the user 'named' with group permissions set to "named" user for security reasons.

- To change the permissions and ownership of the db.openna file, use:

```
[root@deep /]# chmod 644 /var/named/db.openna
[root@deep /]# chown named.named /var/named/db.openna
```

/var/named/78.35.207.in-addr.arpa: Host Names to Addr Mapping

This section applies only if you chose to install and use ISC BIND & DNS as a Primary Name Server in your system. Use this configuration file for the server on your network that acts as a Master Name Server. The "78.35.207.in-addr.arpa" file maps host names to addresses.

Step 1
Create the file in question under your /var/named directory.

- Create the **78.35.207.in-addr.arpa** file (touch /var/named/78.35.207.in-addr.arpa) and add the following lines in the file:

```
;$ORIGIN 78.35.207.in-addr.arpa.
$TTL 172800
@  IN  SOA  ns1.openna.com. root.openna.com.  (
                                        01      ; Serial
                                        10800   ; Refresh after 3 hours
                                        3600    ; Retry after 1 hour
                                        604800  ; Expire after 1 week
                                        172800 ); Minimum TTL of 2 days

; Name Servers (NS) records.
;
                IN      NS      ns1.openna.com.
                IN      NS      ns2.openna.com.

; Addresses Point to Canonical Names (PTR) for Reverse lookups
;
1               IN      PTR     router.openna.com.
2               IN      PTR     gtw.openna.com.
3               IN      PTR     www.openna.com.
4               IN      PTR     smtp.openna.com.
```

Step 2
Now, set the permissions of the 78.35.207.in-addr.arpa file to be (0644/-rw-r--r--) and owned by the user 'named' with group permissions set to "named" user for security reasons.

- To change permissions and ownership of the 78.35.207.in-addr.arpa file, use:

```
[root@deep /]# chmod 644 /var/named/78.35.207.in-addr.arpa
[root@deep /]# chown named.named /var/named/78.35.207.in-addr.arpa
```

Running ISC BIND & DNS as Secondary Slave Name Server

This section applies only if you chose to install and use ISC BIND & DNS as a Secondary Name Server in your system. The purpose of a Slave Name Server is to share the load with the Master Name Server, or handle the entire load if the Master Name Server is down.

A Slave Name Server, which is an authoritative server, loads its data over the network from another Name Server (usually the Master Name Server, but it can load from another Slave Name Server too). This process is called a zone transfer. Slave servers are used to provide necessary redundancy on the network.

/etc/named.conf: The ISC BIND & DNS Configuration File

Use this configuration for the server on your network that acts as a Slave Name Server. You must modify the "named.conf" file on the Slave Name Server host. Text in bold are the parts of the configuration file that change from the previous named.conf file. Don't forget to adjust the values to satisfy your needs.

Step 1
Change every occurrence of primary to secondary except for "0.0.127.in-addr.arpa" and add a master's line with the IP address of the Master Server as shown below.

- Create the **named.conf** file (touch /etc/named.conf) and add the following lines in the file. Below is what I recommend you set.

```
// Authorized source addresses.
acl "trusted" {
        localhost;
        192.168.1.0/24;
        207.35.78.0/24;
};

// Known fake source addresses shouldn't be replied to.
acl "bogon" {
        0.0.0.0/8;
        1.0.0.0/8;
        2.0.0.0/8;
        192.0.2.0/24;
        224.0.0.0/3;
        169.254.0.0/16;
        // Enterprise networks may or may not be bogus.
        10.0.0.0/8;
        172.16.0.0/12;
        192.168.0.0/16;
};

options {
        directory "/var/named";
        allow-transfer { none; };
        allow-query { trusted; };
        allow-recursion { trusted; };
        blackhole { bogon; };
        tcp-clients 1024;
        forwarders { none; };
        version "OpenNA Linux";
};

logging {
        category lame-servers { null; };
};

// Root server hints
zone "." { type hint; file "db.cache"; };

// Provide a reverse mapping for the loopback address 127.0.0.1/24
zone "localhost" {
        type master;
        file "db.localhost";
        notify no;
};

zone "0.0.127.in-addr.arpa" {
        type master;
```

```
        file "0.0.127.in-addr.arpa";
        notify no;
};

// We are the slave server for OpenNA.com
zone "openna.com" {
        type slave;
        file "db.openna";
        masters { 207.35.78.5; };
        allow-query { any; };
};

// Provide a reverse mapping for domains network 207.35.78.0/27
zone "78.35.207.in-addr.arpa" {
        type slave;
        file "78.35.207.in-addr.arpa";
        masters { 207.35.78.5; };
        allow-query { any; };
};
```

The above `named.conf` file tells the Secondary Name Server that it is a Slave Server for the zone "`openna.com`" and should track the version of this zone that is being kept on the host "`207.35.78.5`", which is the Master Name Server in the network.

Step 2
Now, set the permissions of the `named.conf` file to be (`0600/-rw-------`) and owned by the user '`named`' with group permissions set to "`named`" user for security reasons.

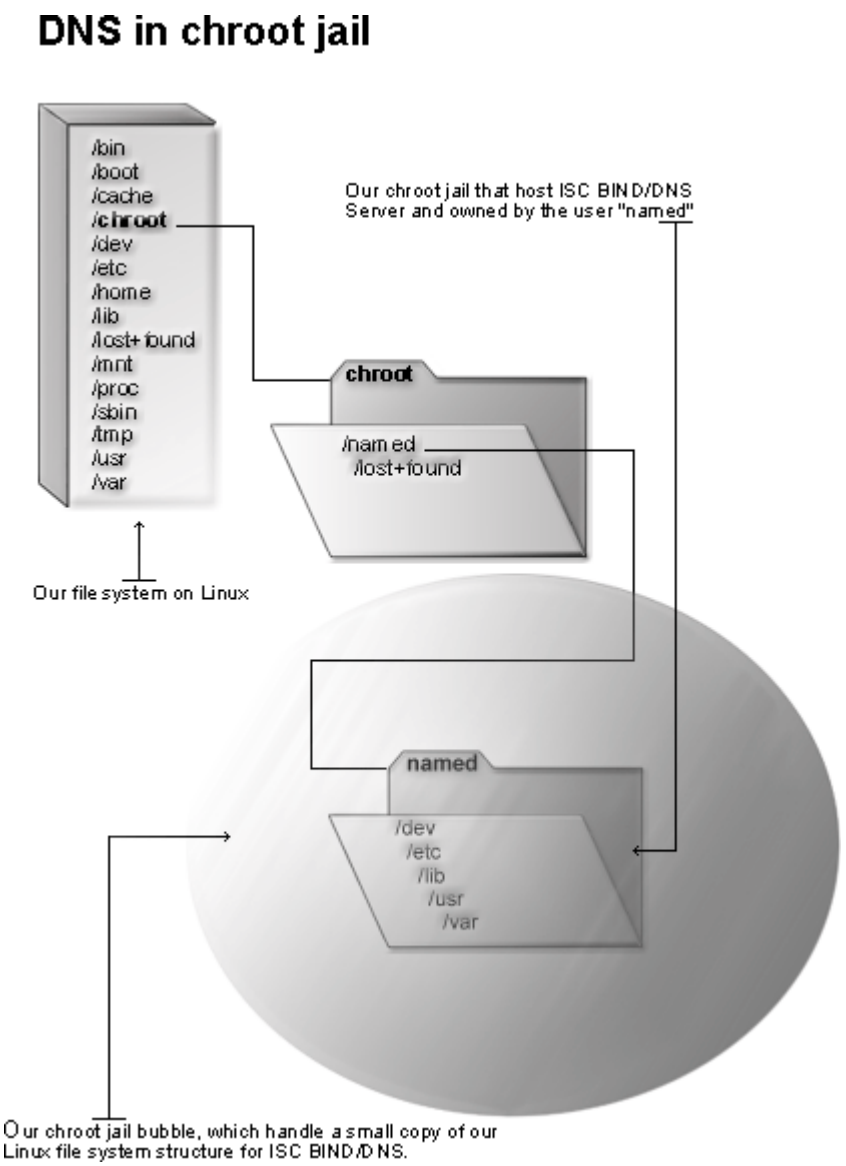
- To change the permissions and ownership of the `named.conf` file, use:
  ```
  [root@deep /]# chmod 600 /etc/named.conf
  [root@deep /]# chown named.named /etc/named.conf
  ```

Running ISC BIND & DNS in a chroot jail

This part focuses on preventing ISC BIND & DNS from being used as a point of break-in to the system hosting it. Since ISC BIND & DNS performs a relatively large and complex function, the potential for bugs that affect security is rather high with this software. In fact, there have been many exploitable bugs in the past that allowed a remote attacker to obtain root access to hosts running ISC BIND & DNS.

To minimize this risk, ISC BIND & DNS can be run **as a non-root user**, which will limit any damage to what can be done as a normal user with a local shell. Of course, this is not enough for the security requirements of most DNS servers, so an additional step can be taken - that is, **running ISC BIND & DNS in a chroot jail.**

The main benefit of a chroot jail is that the jail will limit the portion of the file system the DNS daemon program can see to the root directory of the jail. Additionally, since the jail only needs to support DNS, the programs related to ISC BIND & DNS available in the jail can be extremely limited. More importantly, there is no need for setuid-root programs, which can be used to gain root access and break out of the jail.


DNS in chroot jail
/bin
/boot
/cache
/chroot
/dev
/etc
/home
/lib
/lost+found
/mnt
/proc
/sbin
/tmp
/usr
/var
Our chroot jail that host ISC BIND/DNS
Server and owned by the user "named"
chroot
/named
/lost+found
Our file system on Linux
named
/dev
/etc
/lib
/usr
/var
Our chroot jail bubble, which handle a small copy of our
Linux file system structure for ISC BIND/DNS.

Necessary steps to run ISC BIND & DNS software in a chroot jail:

What you're essentially doing is creating a skeleton root file system with enough of the components necessary (directories, files, etc.) to allow Unix to do a chroot when the ISC BIND & DNS daemon starts. Contrary to its predecessor (Bind8), Bind9 is far easier to setup in a chroot jail environment. Now there is no need to copy the shared library dependencies of the named binary as well as binary programs to the jail. All you have to do is to copy its configuration file with its zone files and instruct its daemon process to chroot to the appropriate chroot directory before starting.

Step 1
The first step in for running ISC BIND & DNS in a chroot jail will be to set up the chroot environment, and create the root directory of the jail. We've chosen /chroot/named for this purpose because we want to put this on its own, separate, file system to prevent file system attacks. Earlier, in our Linux installation procedure, we created a special partition /chroot for this exact purpose.

```
[root@deep /]# /etc/init.d/named stop ← Only if named daemon already run.
Shutting down Named:              [OK]

[root@deep /]# mkdir -p /chroot/named/etc
[root@deep /]# mkdir -p /chroot/named/dev
[root@deep /]# mkdir -p /chroot/named/var/run/named
[root@deep /]# mkdir -p /chroot/named/var/named
[root@deep /]# chown -R named.named /chroot/named/etc/
[root@deep /]# chown -R named.named /chroot/named/dev/
[root@deep /]# chown -R named.named /chroot/named/var/named/
[root@deep /]# chown -R named.named /chroot/named/var/run/named/
```

We need all of the above directories because, from the point of the chroot, we're sitting at "/" and anything above this directory is inaccessible.

Step 2
Next, we must move the main configuration files of ISC BIND & DNS to the appropriate places in the chroot jail and create the random character device. This includes the named.conf file and all zone files.

```
[root@deep /]# mv /etc/named.conf /chroot/named/etc/
[root@deep /]# mv /var/named/* /chroot/named/var/named/
[root@deep /]# mknod /chroot/named/dev/random c 1 8
[root@deep /]# chmod 644 /chroot/named/dev/random
[root@deep /]# chown named.named /chroot/named/etc/named.conf
[root@deep /]# chown -R named.named /chroot/named/var/named/*
```

Step 3
For additional security, we can 'chattr' the named.conf file in the chroot jail directory.

- This procedure can be accomplished with the following commands:

```
[root@deep named]# cd /chroot/named/etc/
[root@deep etc]# chattr +i named.conf
```

WARNING: Don't forget to remove the immutable bit on this file if you have to make some modifications to it later, use the "chattr -i" command.

Step 4
Once the required files to run ISC BIND & DNS are in the chroot jail environment, we can remove the unnecessary directories related to ISC BIND & DNS from the system, since the ones we'll work with now on a daily basis are located under the chroot directory. These directories are /var/named and /var/run/named.

```
[root@deep /]# rm -rf /var/named/
[root@deep /]# rm -rf /var/run/named/
```

Step 5
After that, it is time to instruct ISC BIND & DNS to start in the chroot jail. The /etc/sysconfig/named file is used for this purpose.

- Edit the **named** file (vi /etc/sysconfig/named) and change the following line:

  ```
  #ROOTDIR="/chroot/named/"
  ```

 To read:

  ```
  ROOTDIR="/chroot/named/"
  ```

The "ROOTDIR="/chroot/named/"" option instructs ISC BIND & DNS where the chroot directory is located. Therefore, the named daemon reads this line and chroot's to the specified directory before starting.

Step 6
Finally, we must test the new chrooted jail configuration of our ISC BIND & DNS server.

- Start the new chrooted jail ISC BIND & DNS with the following command:
  ```
  [root@deep /]# /etc/init.d/named start
  Starting Named:                            [OK]
  ```

- If you don't get any errors, do a **ps ax | grep named** and see if we're running:
  ```
  [root@deep /]# ps ax | grep named
  21723 ?        S      0:00 /usr/sbin/named -u named -t /chroot/named/
  ```

If so, lets check to make sure it's chrooted by picking out its process number and doing **ls -la /proc/that_process_number/root/**.

```
[root@deep /]# ls -la /proc/21723/root/
```

If you see something like:

```
total 20
drwxr-xr-x    5 root     root         4096 May 28 12:53 ./
drwxr-xr-x    4 root     root         4096 May 28 12:53 ../
drwxr-xr-x    2 named    named        4096 May 28 12:53 dev/
drwxr-xr-x    2 named    named        4096 May 28 12:54 etc/
drwxr-xr-x    4 root     root         4096 May 28 12:53 var/
```

Congratulations! Your ISC BIND & DNS in a chroot jail is working.

Securing ISC BIND & DNS

This section deals particularly with actions we can take to improve and tighten the security of ISC BIND & DNS. The interesting points here are that we refer to the features available within the base installed program and not to any additional software.

TSIG based transaction security with BIND:

This section applies only if you chose to install and use ISC BIND & DNS as a Master or Slave Name Server on your system. The new BIND9 which is a major rewrite of almost all aspects of the underlying BIND architecture allows us to create transaction keys and use **T**ransaction **SIG**natures (TSIG) with ISC BIND & DNS (TSIG is used for signed DNS requests).

This means that if the server receives a message signed by this key, it can verify the signature. If the signature succeeds, the same key signs the response.

This new feature of BIND will allow us to have better control about who can make a zone transfer, notify, and recursive query messages on the DNS server. It might also be useful for dynamic updates too. Below, we show you the steps to generate this key and how to use it in your named.conf file.

Step 1
The first step will be to generate shared keys for each pair of hosts. This shared secret will be shared between Primary Domain Name Server and Secondary Domain Name Server and an arbitrary key name must be chosen like in our example "ns1-ns2". It is also important that the key name be the same on both hosts.

- To generate shared keys, use the following command:
```
[root@deep /]# dnssec-keygen -a hmac-md5 -b 128 -n HOST ns1-ns2
Kns1-ns2.+157+57071
```

Step 2
The above command will generate a 128 bit (16 byte) HMAC-MD5 key and the result will be in a file called "Kns1-ns2.+157+57071.private" with a base-64 encoded string following the word "Key:", which must be extracted from the file and used as the shared secret.

- Edit the **Kns1-ns2.+157+57071.private** file (vi Kns1-ns2.+157+57071.private), and extract the base-64 encoded string:

```
Private-key-format: v1.2
Algorithm: 157 (HMAC_MD5)
Key: 7Mlb6QwKpGLNzN28zcBm6A==
```

The string "7Mlb6QwKpGLNzN28zcBm6A==" in the above example is the part of this file that must be extracted and used as the shared secret.

Step 3
Once the required base-64 encoded string has been extracted from the generated file, we can remove the files from our system and copy the shared secret to both machines via a secure transport mechanism like ssh, telephone, etc.

- To remove the generated files from the system, use the following commands:
```
[root@deep /]# rm -f Kns1-ns2.+157+57071.key
[root@deep /]# rm -f Kns1-ns2.+157+57071.private
```

Step 4
After that, it is time to inform the servers (Primary & Secondary) of the Key's existence by adding to each server's `named.conf` file the following parameters.

- Edit the **`named.conf`** file (`vi /chroot/named/etc/named.conf`) on both `DNS` servers, and add the following lines:

```
key "ns1-ns2" {
        algorithm hmac-md5;
        secret "7Mlb6QwKpGLNzN28zcBm6A==";
};
```

Once the above lines have been added, your `named.conf` file on both `DNS` servers (Primary & Secondary) should look like:

Under Primary/Master Name Server:

```
// Authorized source addresses.
acl "trusted" {
        localhost;
        192.168.1.0/24;
        207.35.78.0/24;
};

// Known fake source addresses shouldn't be replied to.
acl "bogon" {
        0.0.0.0/8;
        1.0.0.0/8;
        2.0.0.0/8;
        192.0.2.0/24;
        224.0.0.0/3;
        169.254.0.0/16;
        // Enterprise networks may or may not be bogus.
        10.0.0.0/8;
        172.16.0.0/12;
        192.168.0.0/16;
};

key "ns1-ns2" {
        algorithm hmac-md5;
        secret "7Mlb6QwKpGLNzN28zcBm6A==";
};

options {
        directory "/var/named";
        allow-transfer { 207.35.78.6; };
        allow-query { trusted; };
        allow-recursion { trusted; };
        blackhole { bogon; };
        tcp-clients 1024;
        forwarders { none; };
        version "OpenNA Linux";
};

logging {
        category lame-servers { null; };
};

// Root server hints
zone "." { type hint; file "db.cache"; };
```

```
// Provide a reverse mapping for the loopback address 127.0.0.1/24
zone "localhost" {
        type master;
        file "db.localhost";
        notify no;
};

zone "0.0.127.in-addr.arpa" {
        type master;
        file "0.0.127.in-addr.arpa";
        notify no;
};

// We are the master server for OpenNA.com
zone "openna.com" {
        type master;
        file "db.openna";
        allow-query { any; };
};

// Provide a reverse mapping for domains network 207.35.78.0/27
zone "78.35.207.in-addr.arpa" {
        type master;
        file "78.35.207.in-addr.arpa";
        allow-query { any; };
};
```

Under Secondary/Slave Name Server:

```
// Authorized source addresses.
acl "trusted" {
        localhost;
        192.168.1.0/24;
        207.35.78.0/24;
};

// Known fake source addresses shouldn't be replied to.
acl "bogon" {
        0.0.0.0/8;
        1.0.0.0/8;
        2.0.0.0/8;
        192.0.2.0/24;
        224.0.0.0/3;
        169.254.0.0/16;
        // Enterprise networks may or may not be bogus.
        10.0.0.0/8;
        172.16.0.0/12;
        192.168.0.0/16;
};

key "ns1-ns2" {
        algorithm hmac-md5;
        secret "7Mlb6QwKpGLNzN28zcBm6A==";
};

options {
        directory "/var/named";
        allow-transfer { none; };
        allow-query { trusted; };
        allow-recursion { trusted; };
        blackhole { bogon; };
        tcp-clients 1024;
```

```
        forwarders { none; };
        version "OpenNA Linux";
};

logging {
        category lame-servers { null; };
};

// Root server hints
zone "." { type hint; file "db.cache"; };

// Provide a reverse mapping for the loopback address 127.0.0.1/24
zone "localhost" {
        type master;
        file "db.localhost";
        notify no;
};

zone "0.0.127.in-addr.arpa" {
        type master;
        file "0.0.127.in-addr.arpa";
        notify no;
};

// We are the slave server for OpenNA.com
zone "openna.com" {
        type slave;
        file "db.openna";
        masters { 207.35.78.5; };
        allow-query { any; };
};

// Provide a reverse mapping for domains network 207.35.78.0/27
zone "78.35.207.in-addr.arpa" {
        type slave;
        file "78.35.207.in-addr.arpa";
        masters { 207.35.78.5; };
        allow-query { any; };
};
```

Step 5
One of the last steps is to instruct the both servers (Primary & Secondary) to Use the Key. The servers must be told when keys are to be used. Adding another parameter into the `named.conf` file on both `DNS` servers does this. In this parameter, on `ns1` we add the `IP` address of `ns2` and on `ns2` we add the `IP` address of `ns1`.

- Edit the **`named.conf`** file (`vi /chroot/named/etc/named.conf`) on both `DNS` servers, and add the following lines:

```
server x.x.x.x {
        keys { "ns1-ns2"; };
};
```

Where `x.x.x.x` is the `IP` address.

Once the above lines have been added, your `named.conf` file on both `DNS` servers (Primary & Secondary) should look like:

Under Primary/Master Name Server:

```
// Authorized source addresses.
acl "trusted" {
        localhost;
        192.168.1.0/24;
        207.35.78.0/24;
};

// Known fake source addresses shouldn't be replied to.
acl "bogon" {
        0.0.0.0/8;
        1.0.0.0/8;
        2.0.0.0/8;
        192.0.2.0/24;
        224.0.0.0/3;
        169.254.0.0/16;
        // Enterprise networks may or may not be bogus.
        10.0.0.0/8;
        172.16.0.0/12;
        192.168.0.0/16;
};

key "ns1-ns2" {
        algorithm hmac-md5;
        secret "7Mlb6QwKpGLNzN28zcBm6A==";
};

server 207.35.78.6 {
        keys { "ns1-ns2"; };
};

options {
        directory "/var/named";
        allow-transfer { 207.35.78.6; };
        allow-query { trusted; };
        allow-recursion { trusted; };
        blackhole { bogon; };
        tcp-clients 1024;
        forwarders { none; };
        version "OpenNA Linux";
};

logging {
        category lame-servers { null; };
};

// Root server hints
zone "." { type hint; file "db.cache"; };

// Provide a reverse mapping for the loopback address 127.0.0.1/24
zone "localhost" {
        type master;
        file "db.localhost";
        notify no;
};

zone "0.0.127.in-addr.arpa" {
        type master;
        file "0.0.127.in-addr.arpa";
        notify no;
};
```

```
// We are the master server for OpenNA.com
zone "openna.com" {
        type master;
        file "db.openna";
        allow-query { any; };
};

// Provide a reverse mapping for domains network 207.35.78.0/27
zone "78.35.207.in-addr.arpa" {
        type master;
        file "78.35.207.in-addr.arpa";
        allow-query { any; };
};
```

Under Secondary/Slave Name Server:

```
// Authorized source addresses.
acl "trusted" {
        localhost;
        192.168.1.0/24;
        207.35.78.0/24;
};

// Known fake source addresses shouldn't be replied to.
acl "bogon" {
        0.0.0.0/8;
        1.0.0.0/8;
        2.0.0.0/8;
        192.0.2.0/24;
        224.0.0.0/3;
        169.254.0.0/16;
        // Enterprise networks may or may not be bogus.
        10.0.0.0/8;
        172.16.0.0/12;
        192.168.0.0/16;
};

key "ns1-ns2" {
        algorithm hmac-md5;
        secret "7Mlb6QwKpGLNzN28zcBm6A==";
};

server 207.35.78.5 {
        keys { "ns1-ns2"; };
};

options {
        directory "/var/named";
        allow-transfer { none; };
        allow-query { trusted; };
        allow-recursion { trusted; };
        blackhole { bogon; };
        tcp-clients 1024;
        forwarders { none; };
        version "OpenNA Linux";
};

logging {
        category lame-servers { null; };
};

// Root server hints
```

```
zone "." { type hint; file "db.cache"; };

// Provide a reverse mapping for the loopback address 127.0.0.1/24
zone "localhost" {
        type master;
        file "db.localhost";
        notify no;
};

zone "0.0.127.in-addr.arpa" {
        type master;
        file "0.0.127.in-addr.arpa";
        notify no;
};

// We are the slave server for OpenNA.com
zone "openna.com" {
        type slave;
        file "db.openna";
        masters { 207.35.78.5; };
        allow-query { any; };
};

// Provide a reverse mapping for domains network 207.35.78.0/27
zone "78.35.207.in-addr.arpa" {
        type slave;
        file "78.35.207.in-addr.arpa";
        masters { 207.35.78.5; };
        allow-query { any; };
};
```

Step 6
Restart your DNS server on both sides for the changes to take effect.

- Restart ISC BIND & DNS with the following command on both DNS servers:

```
[root@deep /]# /etc/init.d/named restart
Shutting down Named:            [OK]
Starting Named:                 [OK]
```

WARNING: With TSIG feature enabled on your DNS server, it is important to be sure that the clocks on the client and server are synchronized. TSIG includes a timestamp to reduce the potential for replay attacks. If the client and server's clocks are out by too much, TSIG validations will inevitably fail.

Using TSIG key based access control to make a zone transfer:

This section applies only if you chose to install and use ISC BIND & DNS as a Master or Slave Name Server in your system. Once the TSIG feature has been configured and enabled in your DNS server, we can use it to improve security on the system.

One improvement that can be made is with the allow-transfer statement of ISC BIND & DNS. Usually, we configure our Primary/Master Domain Name Server to respond to zone transfers requests from authorized IP addresses only. In most cases, we'll only authorize our known Secondary/Slave Domain Name Servers. The same technique as described here can also be used for dynamic updates, notifications, and recursive query messages.

With BIND9, we do that within a zone phrase in the Primary Name Server with a directive like "allow-transfer { 207.35.78.6; };", but with the sharing of keys between ns1 and ns2 like we previously did, we have extended the possibility of our named.conf file to allow TSIG keys and can use this feature to modify the allow-transfer directive, which will improve security of zone transfer between ns1 and ns2.

- To use TSIG key based access control to make a zone transfer between Primary DNS & Secondary DNS, edit your **named.conf** file on the Primary/Master Domain Name Server ONLY (vi /chroot/named/etc/named.conf) and change the line:

  ```
  allow-transfer { 207.35.78.6; };
  ```

 To Read:

  ```
  allow-transfer { key ns1-ns2; };
  ```

This allows a zone transfer to succeed only if a key named "ns1-ns2" signed the request, which only your Primary & Secondary DNS knows and has in their named.conf. Once the above line has been modified, your named.conf file on the Primary/Master Name Server should look like:

```
// Authorized source addresses.
acl "trusted" {
        localhost;
        192.168.1.0/24;
        207.35.78.0/24;
};

// Known fake source addresses shouldn't be replied to.
acl "bogon" {
        0.0.0.0/8;
        1.0.0.0/8;
        2.0.0.0/8;
        192.0.2.0/24;
        224.0.0.0/3;
        169.254.0.0/16;
        // Enterprise networks may or may not be bogus.
        10.0.0.0/8;
        172.16.0.0/12;
        192.168.0.0/16;
};

key "ns1-ns2" {
        algorithm hmac-md5;
        secret "7Mlb6QwKpGLNzN28zcBm6A==";
};

server 207.35.78.6 {
```

```
        keys { "ns1-ns2"; };
};

options {
        directory "/var/named";
        allow-transfer { key ns1-ns2; };
        allow-query { trusted; };
        allow-recursion { trusted; };
        blackhole { bogon; };
        tcp-clients 1024;
        forwarders { none; };
        version "OpenNA Linux";
};

logging {
        category lame-servers { null; };
};

// Root server hints
zone "." { type hint; file "db.cache"; };

// Provide a reverse mapping for the loopback address 127.0.0.1/24
zone "localhost" {
        type master;
        file "db.localhost";
        notify no;
};

zone "0.0.127.in-addr.arpa" {
        type master;
        file "0.0.127.in-addr.arpa";
        notify no;
};

// We are the master server for OpenNA.com
zone "openna.com" {
        type master;
        file "db.openna";
        allow-query { any; };
};

// Provide a reverse mapping for domains network 207.35.78.0/27
zone "78.35.207.in-addr.arpa" {
        type master;
        file "78.35.207.in-addr.arpa";
        allow-query { any; };
};
```

WARNING: If you use `BIND9`'s dynamic update functionality, you'll also want to restrict zone updates to authorized `IP` addresses and you'd probably do this in the zone phrase. Note that if you don't specify an `allow-update` option, updates are not allowed for that zone so you'll only need to do this if you actually use dynamic update.

```
zone "openna.com" {
        type master;
        file "db.openna";
        allow-update { key ns1-ns2; };
        allow-query { any; };
};
```

Using encryption algorithm for the name server control utility `rndc`:

This section applies to all types of `ISC BIND & DNS`. The `BIND9` utility for controlling the name server, `rndc`, has its own configuration file `/etc/rndc.conf`, which also requires a `TSIG` key to work. The name server must be configured to accept `rndc` connections and to recognize the key specified in the `rndc.conf` file, using the `controls` statement in `named.conf`. Below are the procedures to follow before using `rndc` on your system.

Step 1
The first step will be to generate shared keys. This shared secret key will be included into `/etc/rndc.conf` file and `/chroot/named/etc/named.conf` file.

- To generate a random shared key, use the following command:
  ```
  [root@deep /]# dnssec-keygen -a hmac-md5 -b 128 -n user rndc
  Krndc.+157+22629
  ```

Step 2
The above command will generate a 128 bit (16 byte) `HMAC-MD5` key and the result will be in a file called "`Krndc.+157+22629.private`" with a base-64 encoded string following the word "`Key:`", which must be extracted from the file and used as the shared secret.

- Edit the **`Krndc.+157+22629.private`** file (vi `Krndc.+157+22629.private`), and extract the base-64 encoded string:

  ```
  Private-key-format: v1.2
  Algorithm: 157 (HMAC_MD5)
  Key: eRKnIU6WhEWB7XGmvXexrA==
  ```

The string "`eRKnIU6WhEWB7XGmvXexrA==`" in the above example is the part of this file that must be extracted and used as the shared secret.

Step 3
Once the required base-64 encoded string has been extracted from the generated file, we can remove the files from our system and copy the shared secret to both the `rndc.conf` and `named.conf` files.

- To remove the generated files from the system, use the following commands:
  ```
  [root@deep /]# rm -f Krndc.+157+22629.key
  [root@deep /]# rm -f Krndc.+157+22629.private
  ```

Step 4
After that, we must edit the `rndc.conf` file and configure it with the key.

- Edit the **`rndc.conf`** file (`vi /etc/rndc.conf`), and add the following lines:

  ```
  options {
          default-server  localhost;
          default-key     "key";
  };

  server localhost {
          key     "key";
  };
  ```

```
key "key" {
        algorithm       hmac-md5;
        secret "eRKnIU6WhEWB7XGmvXexrA==";
};
```

In the above example, `rndc` will by default use the server at localhost (`127.0.0.1`) and the key called "`key`". Commands to the localhost server will use the "`key`" key. The key statement indicates that "`key`" uses the `HMAC-MD5` algorithm and its secret clause contains the base-64 encoding of the `HMAC-MD5` secret enclosed in double quotes.

Step 5
Also don't forget to edit the `named.conf` file and configure it with the key.

- Edit the **`named.conf`** file (`vi /chroot/named/etc/named.conf`), and add the lines:

```
// Authorized source addresses.
acl "trusted" {
        localhost;
        192.168.1.0/24;
        207.35.78.0/24;
};

// Known fake source addresses shouldn't be replied to.
acl "bogon" {
        0.0.0.0/8;
        1.0.0.0/8;
        2.0.0.0/8;
        192.0.2.0/24;
        224.0.0.0/3;
        169.254.0.0/16;
        // Enterprise networks may or may not be bogus.
        10.0.0.0/8;
        172.16.0.0/12;
        192.168.0.0/16;
};

key "key" {
        algorithm hmac-md5;
        secret "eRKnIU6WhEWB7XGmvXexrA==";
};

controls {
        inet 127.0.0.1 allow { localhost; } keys { "key"; };
};

key "ns1-ns2" {
        algorithm hmac-md5;
        secret "7Mlb6QwKpGLNzN28zcBm6A==";
};

server 207.35.78.6 {
        keys { "ns1-ns2"; };
};

options {
        directory "/var/named";
        allow-transfer { key ns1-ns2; };
        allow-query { trusted; };
        allow-recursion { trusted; };
        blackhole { bogon; };
```

```
        tcp-clients 1024;
        forwarders { none; };
        version "OpenNA Linux";
};

logging {
        category lame-servers { null; };
};

// Root server hints
zone "." { type hint; file "db.cache"; };

// Provide a reverse mapping for the loopback address 127.0.0.1/24
zone "localhost" {
        type master;
        file "db.localhost";
        notify no;
};

zone "0.0.127.in-addr.arpa" {
        type master;
        file "0.0.127.in-addr.arpa";
        notify no;
};

// We are the master server for OpenNA.com
zone "openna.com" {
        type master;
        file "db.openna";
        allow-query { any; };
};

// Provide a reverse mapping for domains network 207.35.78.0/27
zone "78.35.207.in-addr.arpa" {
        type master;
        file "78.35.207.in-addr.arpa";
        allow-query { any; };
};
```

In the above example, `rndc` connections will only be accepted at localhost. Don't forget to integrate this security feature into all types of `DNS` servers you may have on your network. The above information works with Caching-Only, Primary and Secondary Name Servers.

Step 6
Finally, it is important to restart your `DNS` server for the changes to take effect.

- Restart `ISC BIND & DNS` with the following command:

```
[root@deep /]# /etc/init.d/named restart
Shutting down Named:           [OK]
Starting Named:                [OK]
```

WARNING: Using the encryption algorithm for the name server control utility `rndc` doesn't work with the `lwresd.conf` file. It only works with `named.conf` file and not with `lwresd.conf`.

DNSSEC Cryptographic authentication of DNS information:

This section applies only if you chose to install and use ISC BIND & DNS as a Master or Slave Name Server on your system. The BIND9 release of ISC BIND & DNS includes and support validation of DNSSEC (**DNS Sec**urity) signatures in responses but should still be considered experimental. The DNSSEC feature of BIND9 is used for signed zones, what DNSSEC does is to make sure that the DNS communication taking place is with the correct server, and that the information has not been tampered with during transport. This allows protection of Internet-wide DNS transfers, cache pollution, and will protect you from someone trying to spoof your DNS servers.

But be aware that DNSSEC is NOT for all types of Name Server. DNSSEC verifies that the data received by a resolver is the same as the published data. For it to do anything, your resolver must be configured to verify data. Signing a localhost zone like for Caching-Only or Secondary/Slave Name Server is not useful, since it's not traveling over an insecure network. Signing data in general doesn't help you; it just guarantees that anyone that gets data from your server can verify its correctness, if they've configured their resolver to do so.

Each zone (domain) in the DNS will need to have a key pair. The zone's public key will be included in its resource records. The zone's private key will be kept securely by the administrator of the zone, and never given to anyone outside your organization. Below, I show you steps for the creation and the use of DNSSEC signed zones.

In our example we assume that you want to use the DNSSEC feature for your Primary/Master Name Server with your parent zone (i.e. .COM) over the Internet. All commands listed below are assumed to be made in the **/chroot/named/var/named** directory since the DNSSEC tools require that the generated key files will be in the working directory.

Step 1
As usual in the cryptography area, the first step will be to generate a key pair. The generated zone keys here will produce a private and public key that will be used to sign records for the related zones in question as usual the zone keys must have the same name as the zone like in our example "openna.com". The resulting public keys should later be inserted into the related zone file with the **$INCLUDE** statements.

- To generate a 1024 bit DSA key for the openna.com zone, use the following commands:
```
[root@deep /]# cd /chroot/named/var/named/
[root@deep named]# dnssec-keygen -a DSA -b 1024 -n ZONE openna.com
Kopenna.com.+003+28448
```

The above command will generate a 1024 bit DSA key for the openna.com zone and two output files will be produced: “Kopenna.com.+003+28448.key” and “Kopenna.com.+003+28448.private”. The private key will be used to generate signatures, and the public key will be used for signature verification.

Step 2
Once the zone keys have been generated as shown previously, a keyset must be built and transmitted to the administrator of the parent zone in question to sign the keys with its own zone key. It is important that when building a keyset, the following information at least be included in the generation of the key: the TTL (**T**ime **T**o **L**ive) of the keyset must be specified, and the desired signature validity period of the parent's signature may also be specified.

- To generate a keyset containing the previous key, use the following command:
```
[root@deep named]# dnssec-makekeyset -t 3600 -e +864000 \
Kopenna.com.+003+28448
keyset-openna.com.
```

The above command generates a keyset containing the previous key with a TTL of 3600 and a signature validity period of 10 days (864000) starting from now to an output file called "keyset-openna.com.". This file should be transmitted to the parent to be signed. It includes the keys, as well as signatures over the keyset generated by the zone keys themselves, which are used to prove ownership of the private keys and encode the desired validity period.

Step 3
After that, the administrator on the parent zone (in our case .COM since our zone is openna.com) should receive the keyset files for each of your secure zones (in our example: keyset-openna.com.) and must sign the keys with its own private key. This is the step that permits others on the net to determine that the resource records that they receive from your zone are really from you.

- The administrator of your parent zone will sign the keyset with its zone keys by using something like the following command:
```
[root@internic named]# dnssec-signkey keyset-openna.com. \
KA.COM.+003+31877
signedkey-openna.com.
```

One output file called "signedkey-openna.com." will be produced. This file should be both transmitted back to the recipient and retained. It will include all keys from the keyset file and signatures generated by this zone's zone keys.

WARNING: Take a note that in our example "KA.COM.+003+31877" is the key for the "A.COM" zone file, which is our parent zone. Olafur Gudmundsson <ogud@ogud.com> has informed me that .COM is not there yet, but what you SHOULD do is to contact your registrar and notify them that you MUST have your key set signed by .COM ASAP and when they expect that to happen. Verisign Global Registry has indicated that they want to start signing .COM sometime this year, but check with them what the current plans are.

To summarize our procedures:

- ✓ We have generated a key pair for our zone file in step 1.
- ✓ We have build and transmitted a key set to our parent zone for signing in step 2.
- ✓ Administrator in the parent zone signs our keyset with his private key.
- ✓ Administrator in the parent zone transmits our key set back after singing it.

Step 4
Ok, from now if we recall what we said before is that the public keys should be inserted into the related zone file with the **$INCLUDE** statements, then at this step, we must insert the public key (`Kopenna.com.+003+28448.key`) into the related zone file, which in our example the zone file called `db.openna` located under `/chroot/named/var/named` directory.

- Edit the **db.openna** zone file (`vi /chroot/named/var/named/db.openna`), and add the following line to your default zone file:

```
;$ORIGIN openna.com.
$TTL 172800
@  IN  SOA  ns1.openna.com. root.openna.com.  (
                                        01      ; Serial
                                        10800   ; Refresh after 3 hours
                                        3600    ; Retry after 1 hour
                                        604800  ; Expire after 1 week
                                        172800 ); Minimum TTL of 2 days

$INCLUDE Kopenna.com.+003+28448.key

; Name Servers (NS) records.
;
        IN      NS      ns1.openna.com.
        IN      NS      ns2.openna.com.

; Mail Exchange (MX) records.
;
        MX      0       smtp.openna.com.

; Addresses for the canonical names (A) records.
;
localhost       IN      A       127.0.0.1
router          IN      A       207.35.78.1
gtw             IN      A       207.35.78.2
www             IN      A       207.35.78.3
smtp            IN      A       207.35.78.4
```

Don't forget to restart your `DNS` server for the change to take effect.

- Restart `ISC BIND & DNS` with the following command:

```
[root@deep /]# /etc/init.d/named restart
Shutting down Named:           [OK]
Starting Named:                [OK]
```

NOTE: Please, check that everything looks right in your log files (`/var/log/messages`) before continuing with the step below. It is important to be sure that there is nothing wrong with your configuration.

Step 5
Once the keyset has been signed and approved by the parent zone (.COM), the final step will be to sign our zone. The result will produce an output file called "db.openna.signed". This file should be referenced by named.conf as the input file for the zone instead of the default one called "db.openna".

- To sign the zone file, use the following command:
```
[root@deep named]# dnssec-signzone -o openna.com db.openna
db.openna.signed
```

NOTE: If a zone doesn't publish a key, then BIND will accept any plausible-looking records, without a digital signature, just like in the original DNS. This provides compatibility with existing DNS zones, allowing Secure DNS to be gradually introduced throughout the Internet.

Step 6
The result of signing the zone will produce an output file called "db.openna.signed". Recall that this file should be referenced by named.conf as the input file for the zone.

- Edit the **named.conf** file (vi /chroot/named/etc/named.conf), and change the following line:

```
// Authorized source addresses.
acl "trusted" {
        localhost;
        192.168.1.0/24;
        207.35.78.0/24;
};

// Known fake source addresses shouldn't be replied to.
acl "bogon" {
        0.0.0.0/8;
        1.0.0.0/8;
        2.0.0.0/8;
        192.0.2.0/24;
        224.0.0.0/3;
        169.254.0.0/16;
        // Enterprise networks may or may not be bogus.
        10.0.0.0/8;
        172.16.0.0/12;
        192.168.0.0/16;
};

key "key" {
        algorithm hmac-md5;
        secret "eRKnIU6WhEWB7XGmvXexrA==";
};

controls {
        inet 127.0.0.1 allow { localhost; } keys { "key"; };
};

key "ns1-ns2" {
        algorithm hmac-md5;
        secret "7Mlb6QwKpGLNzN28zcBm6A==";
};
```

```
server 207.35.78.6 {
        keys { "ns1-ns2"; };
};

options {
        directory "/var/named";
        allow-transfer { key ns1-ns2; };
        allow-query { trusted; };
        allow-recursion { trusted; };
        blackhole { bogon; };
        tcp-clients 1024;
        forwarders { none; };
        version "OpenNA Linux";
};

logging {
        category lame-servers { null; };
};

// Root server hints
zone "." { type hint; file "db.cache"; };

// Provide a reverse mapping for the loopback address 127.0.0.1/24
zone "localhost" {
        type master;
        file "db.localhost";
        notify no;
};

zone "0.0.127.in-addr.arpa" {
        type master;
        file "0.0.127.in-addr.arpa";
        notify no;
};

// We are the master server for OpenNA.com
zone "openna.com" {
        type master;
        file "db.openna.signed";
        allow-query { any; };
};

// Provide a reverse mapping for domains network 207.35.78.0/27
zone "78.35.207.in-addr.arpa" {
        type master;
        file "78.35.207.in-addr.arpa";
        allow-query { any; };
};
```

Step 7
Restart ISC BIND & DNS using the following command on both DNS servers.

- Restart ISC BIND & DNS with the following command on both DNS servers:

```
[root@deep /]# /etc/init.d/named restart
Shutting down Named:          [OK]
Starting Named:               [OK]
```

Optimizing ISC BIND & DNS

This section deals with actions we can make to further improve and tighten performance of ISC BIND & DNS. Note that we refer to the features available within the base installed program.

The BIND9 Lightweight Resolver:

The new release of BIND comes with a new daemon program called "lwresd". The lwresd daemon is essentially a Caching-Only Name Server that answers requests using the lightweight resolver protocol rather than the DNS protocol. Because it needs to run on each host, it is designed to require no or minimal configuration. In our configuration we'll run lwresd in a chrooted environment.

On all Caching-Only Name Servers that you may have in your network, it can be interesting to run this daemon "lwresd" instead of the full "named" daemon. If we remember that a Caching-Only Name Server is not authoritative for any domains except 0.0.127.in-addr.arpa.

It can look up names inside and outside your zone, as can Primary and Slave Name Servers but the difference is that when it initially looks up a name within your zone, it ends up asking one of the Primary or Slave Names Servers for your zone for the answer and nothing else. Therefore we can run the "lwresd" daemon on this kind of Name Server and everything will run as we want.

Below, are the steps to run your Caching-Only Name Server with the "lwresd" daemon instead of the "named" daemon in a chrooted environment.

Step 1
By default, the lwresd daemon listens on the loopback address (127.0.0.1). With a firewall on the system it is important to instruct the lwresd daemon to listen to the external interface of the server. This can be made with an "lwserver" statement line in the /etc/resolv.conf file.

- Edit the **resolv.conf** file (vi /etc/resolv.conf), and add the following line:

```
lwserver 207.35.78.2
```

Where 207.35.78.2 is the IP address of the external interface in the firewall script file.

Step 2
Since lwresd will run in a chroot jail environment, we must copy the /etc/resolv.conf file to our chrooted environment for the lwresd daemon to be able to find the resolv.conf file and start.

- To copy the resolv.conf file to your chroot jail, use the following command:
 `[root@deep /]# cp /etc/resolv.conf /chroot/named/etc/`

Step 3
Now, we must create an initialization script file for the `lwresd` daemon to automatically start and stop on your server.

- Create **`lwresd`** script file (`touch /etc/init.d/lwresd`) and add the following lines:

```
#!/bin/bash

# This shell script takes care of starting and stopping lwresd.
#
# chkconfig: - 55 45
# description: Lwresd is essentially a Caching-Only Name Server that \
#              answers requests using the lightweight resolver protocol \
#              rather than the DNS protocol.
#
# processname: lwresd

# Source function library.
. /etc/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Source for additional options if we have them.
if [ -f /etc/sysconfig/named ] ; then
        . /etc/sysconfig/named
fi

# Check that networking is up.
[ "${NETWORKING}" = "no" ] && exit 0

# If Lwresd is not available stop now.
[ -f /usr/sbin/lwresd ] || exit 0
[ -f "${ROOTDIR}"/etc/lwresd.conf ] || exit 0
[ -f "${ROOTDIR}"/etc/resolv.conf ] || exit 0

# Path to the Lwresd binary.
lwresd=/usr/sbin/lwresd

RETVAL=0
prog="Lwresd"

start() {
        echo -n $"Starting $prog: "
        if [ -n "${ROOTDIR}" -a "x${ROOTDIR}" != "x/" ]; then
                OPTIONS="${OPTIONS} -t ${ROOTDIR}"
        fi
        daemon $lwresd -P 53 -u named ${OPTIONS}
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/lwresd
        return $RETVAL
}

stop() {
        echo -n $"Shutting down $prog: "
        killproc $lwresd
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/lwresd
        return $RETVAL
}
```

```
# See how we were called.
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  status)
        status $lwresd
        RETVAL=$?
        ;;
  restart)
        stop
        start
        RETVAL=$?
        ;;
  condrestart)
        if [ -f /var/lock/subsys/lwresd ]; then
                stop
                start
                RETVAL=$?
        fi
        ;;
  *)
        echo $"Usage: $0 {start|stop|status|restart|condrestart}"
        exit 1
esac
exit $RETVAL
```

Step 4
Once the `lwresd` script file has been created, it is important to make it executable, change its default permissions, create the necessary links and start it. Making this file executable will allow the system to run it, changing its default permissions is to allow only the root user to change this file for security reasons, and the creation of the symbolic links will let the process control initialization of Linux start the program automatically for you at each boot.

- To make this script executable and to change its default permissions, use the commands:
  ```
  [root@deep /]# chmod 700 /etc/init.d/lwresd
  [root@deep /]# chown 0.0 /etc/init.d/lwresd
  ```

- To create the symbolic `rc.d` links for `lwresd`, use the following commands:
  ```
  [root@deep /]# chkconfig --add lwresd
  [root@deep /]# chkconfig --level 2345 lwresd on
  ```

Step 5
Because we run `lwresd` instead of the `named` daemon in our Caching-Only Name Server, it is important to deactivate and uninstall the `named` initialization script file on our system.

- These procedures can be accomplished with the following commands:
  ```
  [root@deep /]# chkconfig --del named
  [root@deep /]# chkconfig --level 2345 named off
  [root@deep /]# rm -f /etc/init.d/named
  ```

Step 6
The lwresd daemon reads its configuration file from /etc/lwresd.conf. This file is optional and the program can run without it using just the resolv.conf file, but it is preferable to create and use this configuration file with lwresd to reduce the possibility of messages in the log file.

The format of lwresd.conf file is identical to named.conf. Therefore all you have to do is to rename your existing Caching-Only Name Server named.conf file to become lwresd.conf file.

- This procedure can be accomplished with the following command:
```
[root@deep /]# cd /chroot/named/etc/
[root@deep etc]# mv named.conf lwresd.conf
```

Step 7
Now it is time to start your DNS server with the lwresd daemon.

- To start lwressd manually, use the following command:
```
[root@deep /]# /etc/init.d/lwresd start
Starting Lwresd:          [OK]
```

ISC BIND & DNS Administrative Tools

The commands listed below are some that we use often, but many more exist. Check the manual pages of ISC BIND & DNS and documentation for more information.

dig

The dig command DNS lookup utility (**d**omain **i**nformation **g**roper) is a tool for interrogating DNS name servers by performing DNS lookups and displays the answers that are returned from. It can also be used to update your db.cache file by telling your server where the servers for the “root” zone are. Dig is a useful tool to use when you need to troubleshoot DNS problems.

- Use the following command to query an address:
```
[root@deep /]# dig @ns1.openna.com

; <<>> DiG 9.2.1 <<>> @ns1.openna.com
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 20066
;; flags: qr rd ra; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 4

;; QUESTION SECTION:
;.                              IN      NS

;; ANSWER SECTION:
.                       87686   IN      NS      C.ROOT-SERVERS.NET.
.                       87686   IN      NS      D.ROOT-SERVERS.NET.
.                       87686   IN      NS      E.ROOT-SERVERS.NET.
.                       87686   IN      NS      F.ROOT-SERVERS.NET.
.                       87686   IN      NS      G.ROOT-SERVERS.NET.
.                       87686   IN      NS      H.ROOT-SERVERS.NET.
.                       87686   IN      NS      I.ROOT-SERVERS.NET.
.                       87686   IN      NS      J.ROOT-SERVERS.NET.
.                       87686   IN      NS      K.ROOT-SERVERS.NET.
.                       87686   IN      NS      L.ROOT-SERVERS.NET.
.                       87686   IN      NS      M.ROOT-SERVERS.NET.
.                       87686   IN      NS      A.ROOT-SERVERS.NET.
.                       87686   IN      NS      B.ROOT-SERVERS.NET.
```

```
;; ADDITIONAL SECTION:
J.ROOT-SERVERS.NET.     174086  IN      A       198.41.0.10
K.ROOT-SERVERS.NET.     174086  IN      A       193.0.14.129
L.ROOT-SERVERS.NET.     174086  IN      A       198.32.64.12
M.ROOT-SERVERS.NET.     174086  IN      A       202.12.27.33

;; Query time: 3 msec
;; SERVER: 207.35.78.5#53(ns1.openna.com)
;; WHEN: Tue May 28 16:24:02 2002
;; MSG SIZE  rcvd: 292
```

Where **@ns1.openna.com** is the address of the server. Many options exist for this tool and I recommend that you read the `dig` manual page `dig(1)` for a complete listing.

rndc

The `rndc` command utility allows the system administrator to control the operation of a name server. It replaces the `ndc(8)` utility that was provided in the old `BIND8` releases. You can use this tool to reload configuration files and zones, schedule immediate maintenance for a zone, write server statistics, toggle query logging, stop the `DNS` server, and many other useful functions. The `rndc` tool prints a short summary of the supported commands and the available options if invoked on command line without options.

- Type `rndc` on your terminal to get a short summary of all available commands:

```
[root@deep /]# rndc
Usage: rndc [-c config] [-s server] [-p port]
        [-k key-file ] [-y key] [-V] command

command is one of the following:

  reload        Reload configuration file and zones.
  reload zone [class [view]]
                Reload a single zone.
  refresh zone [class [view]]
                Schedule immediate maintenance for a zone.
  reconfig      Reload configuration file and new zones only.
  stats         Write server statistics to the statistics file.
  querylog      Toggle query logging.
  dumpdb        Dump cache(s) to the dump file (named_dump.db).
  stop          Save pending updates to master files and stop the server.
  halt          Stop the server without saving pending updates.
  trace         Increment debugging level by one.
  trace level   Change the debugging level.
  notrace       Set debugging level to 0.
  flush         Flushes all of the server's caches.
  flush [view]  Flushes the server's cache for a view.
  status        Display status of the server.
  *restart      Restart the server.

* == not yet implemented
Version: 9.2.1
```

ISC BIND & DNS Users Tools

The commands listed below are some that we often use, but many more exist. Check the manual pages of ISC BIND & DNS and documentation for more information.

nslookup

The nslookup program allows the user to query Internet domain name servers interactively or non-interactively. In interactive mode the user can query name servers for information about various hosts and domains, and print a list of hosts in a domain. In non-interactive mode the user can just print the requested information for a host or domain.

Interactive mode has a lot of options and commands; it is recommended that you check the manual page for nslookup.

- To enter under nslookup Interactive mode, use the command:

```
[root@deep /]# nslookup -sil
> www.openna.com
Server:         207.35.78.5
Address:        207.35.78.5#53

Name:   www.openna.com
Address: 207.35.78.3
> exit
```

- To run in non-interactive mode, use the command:

```
[root@deep /]# nslookup -sil www.openna.com
Server:         207.35.78.5
Address:        207.35.78.5#53

Name:   www.openna.com
Address: 207.35.78.3
```

Where <www.openna.com> is the host name or Internet address of the name server to be looked up.

host

The host tool is a simple utility for performing DNS lookups. It is normally used to convert names to IP addresses and vice versa. When no arguments or options are given, host prints a short summary of its command line arguments and options.

- To print host command line arguments and options, use the command:

```
[root@deep /]# host
Usage: host [-aCdlrTwv] [-c class] [-n] [-N ndots] [-t type] [-W time]
            [-R number] hostname [server]
       -a is equivalent to -v -t *
       -c specifies query class for non-IN data
       -C compares SOA records on authoritative nameservers
       -d is equivalent to -v
       -l lists all hosts in a domain, using AXFR
       -n Use the nibble form of IPv6 reverse lookup
       -N changes the number of dots allowed before root lookup is done
       -r disables recursive processing
       -R specifies number of retries for UDP packets
       -t specifies the query type
       -T enables TCP/IP mode
       -v enables verbose output
       -w specifies to wait forever for a reply
       -W specifies how long to wait for a reply
```

- To look up host names using the domain server, use the command:
```
[root@deep /]# host openna.com
openna.com. has address 207.35.78.3
```

Further documentation

For more details, there are several manual pages related to `BIND` that you could read.

```
$ man named-checkconf (1)        - Configuration file syntax checking tool.
$ man named-checkzone (1)        - Zone validity checking tool.
$ man host (1)                   - DNS lookup utility.
$ man dig (1)                    - DNS lookup utility.
$ man rndc.conf (5)              - rndc configuration file.
$ man named (8)                  - Internet domain name server.
$ man rndc (8)                   - name server control utility.
$ man lwresd (8)                 - lightweight resolver daemon.
$ man nsupdate (8)               - Dynamic DNS update utility.
```

CHAPTER 29

ISC DHCP

IN THIS CHAPTER

1. **Building a kernel with `ISC DHCP` support**
2. **Compiling - Optimizing & Installing `ISC DHCP`**
3. **Configuring `ISC DHCP`**
4. **Testing the `DHCP` server**
5. **Running `ISC DHCP` in a chroot jail**
6. **Securing `ISC DHCP`**
7. **Running the `DHCP` client for Linux**

Linux ISC DHCP

Abstract

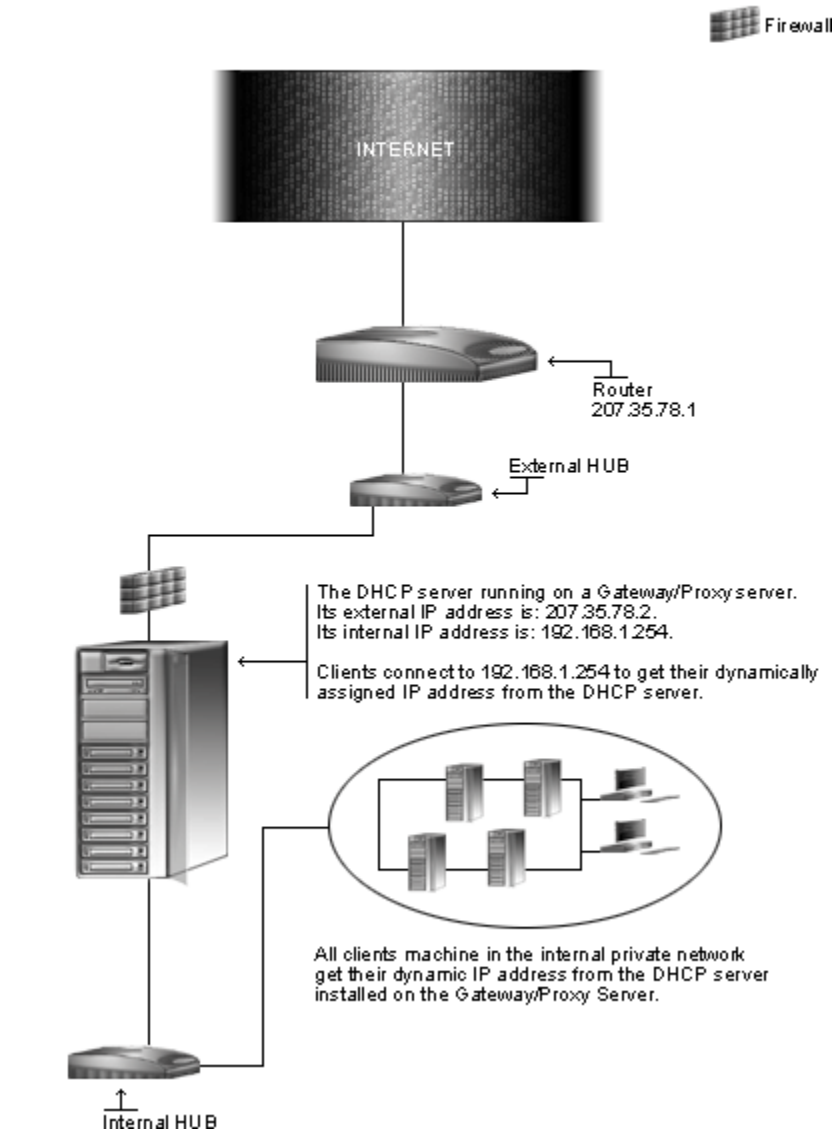
On a network environment where there are more than a few machines to administer, it can become hard and consume a lot of time for an administrator to have to assign a new IP address with Broadcast, Netmask, and Network information, each time a new computer is added to the company network. The situation can become more complicated for the administrator if the IP address range and network information of all systems in the company need to be changed for a new range of IP or network addresses.

There can be many situations in which this would happen, for example, if the company/individual decides for any reason to change its ISP, or networking policies. Other possibilities exist, and it becomes clear that we have to find a way to facilitate the administration and management of IP addresses and other network information in these situations.

DHCP (**D**ynamic **H**ost **C**onfiguration **P**rotocol) is the answer to our problem. It eliminates the need to run around typing in all those IP and Ethernet addresses each time you add a workstation to your existing network. It works well if you have to manage a lot of workstations and also mobile users. Once you configure your computers to use DHCP, the DHCP server will automatically look up an available address and assigns it to the client.

The ISC DHCP software that we will discuss in this chapter provides a DHCP server, DHCP client, as well as a DHCP relay agent. You don't need to install all of these services, but only the ones that you expect to use. The DHCP server is what we use to provide DHCP services to our client computers. The DHCP client is what Linux workstations or servers (sometimes) use to obtain network information from a remote DHCP server and the DHCP relay agent allows you to have a central DHCP server managing more than one subnet.

In this chapter, we will show you how to install, secure, optimize and configure a DHCP server for Linux. We will also discuss the DHCP client for Linux but we will not cover the DHCP relay agent, since this feature is only required for specific situations and it's also poses security risks for our DHCP server and network.



In the above diagram, you can see that client machines go through the Gateway/Proxy server to access the Internet and our `DHCP` server, which dynamically assigns IP addresses to all allowed workstations on the private network where our Gateway/Proxy server resides. This allows us to simplify the administration tasks and to control all external access coming from the private internal network.

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, at personal discretion).
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: Yes
Latest ISC DHCP version number is 3.0p1

The procedures given in this chapter are likely to work on all Linux platforms, but we have only tested it on OpenNA Linux and Red Hat Linux.

Packages

The following is based on information listed by the ISC for DHCP as of 2002/05/08. Please regularly check www.isc.org for the latest status. We chose to install the required component from a source file because it provides the facility to fine tune the installation.

Source code is available from:
ISC DHCP Homepage: http://www.isc.org/
ISC DHCP FTP Site: 204.152.184.27
You must be sure to download: dhcp-3.0p1.tar.gz

Prerequisites

ICS DHCP requires that the software below is already installed on your system to be able to compile successfully. If this is not the case, you must install it. Please make sure you have this program installed on your machine before you proceed with this chapter.

- ✓ ISC BIND & DNS is required to set up ISC DHCP in your system.
- ✓ MAKEDEV is required to build ISC DHCP in your system

NOTE: For more information on ISC BIND & DNS software, see its related chapter in this book.

Building a kernel with ISC DHCP support

The first thing you need to do before going into the ISC DHCP installation and configuration is to ensure that your kernel has been built with "Packet socket" and "Socket Filtering" support. This means you need kernel 2.4.x and answer "**y**" or "**m**" to the following questions depending on the kernel type you have configured.

For Monolithic Kernel, you will answer by "y" and for a Modularized Kernel, you will answer "m". It is important to understand that if "Packet socket" and "Socket Filtering" support are not enabled in your Kernel, NONE of the information contained in this chapter will work.

If your Kernel is one that comes directly from your Linux vendor or is unmodified, then there is a good chance that your kernel is already built with "Packet socket" and "Socket Filtering" support enabled, therefore you don't have to recompile it and/or perform the steps below.

```
* Networking options
*
Packet socket (CONFIG_PACKET) ← Answer Y here
Socket Filtering (CONFIG_FILTER) ← Answer Y here
```

In the above examples, we answer the questions "y" since we assume that you want a Monolithic Kernel, which is faster than a Modularized Kernel. If you prefer to run a Modularized Kernel, you'll have to answer "m" to the questions.

WARNING: If you have followed the Linux Kernel chapter and have recompiled your Kernel, none of the required options for "Packet socket" and "Socket Filtering" support, as shown above, are already set. You have to recompile your kernel and enable support for the above options. It is a good idea to run ISC DHCP service on a Gateway/Proxy Server, which have at least, two network interfaces installed on it. This means that you have to use the required kernel setups for a Gateway/Proxy Server as explained into the GIPTables Firewall chapter of this book.

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all the files installed on the system in the eventuality of an update in the future. To solve the problem, it is a good idea to make a list of files on the system before you install ISC DHCP, and then one afterwards, and then compare them using the **diff** utility to find out what files were placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep root]# find /* > DHCP1
  ```
- And the following one after you install the software:
  ```
  [root@deep root]# find /* > DHCP2
  ```
- Then use the following command to get a list of what changed:
  ```
  [root@deep root]# diff DHCP1 DHCP2 > ISC-DHCP-Installed
  ```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. In the example above, we use the /root directory of the system to store all generated list files.

Compiling - Optimizing & Installing ISC DHCP

Below are the steps that you must make to configure, compile and optimize the ISC DHCP software before installing it onto your system. First off, we install the program as the user 'root' so as to avoid permissioning problems.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- This can be done with the following commands:
  ```
  [root@deep /]# cp dhcp-version.tar.gz /var/tmp/
  [root@deep /]# cd /var/tmp/
  [root@deep tmp]# tar xzpf dhcp-version.tar.gz
  ```

Step 2
In order to check that the version of ISC DHCP, which you are going to install, is an original and unmodified one, please check the supplied signature with the PGP key from ISC DHCP. Unfortunately, ISC DHCP doesn't provide a MD5 signature for verification. But a PGP key is available on the ISC DHCP website.

To get a PGP key copy of ISC DHCP, please point your browser to the following URL: http://www.isc.org/. For more information about how to use this key for verification, see the GnuPG chapter in this book.

Step 3
After that, move into the newly created ISC DHCP directory and perform the following steps before compiling and optimizing it. The modifications we bring to the ISC DHCP source files below are necessary to relocate some of the default files, make the DHCP software server run faster depending of the number of client's machines that you expect to run with it, as well as to be compatible with our Linux operating system.

- To move into the newly created ISC DHCP directory, use the following command:
 `[root@deep tmp]# cd dhcp-3.0p1/`

Step 4
The first file that we must modify is called **site.conf** located in the source directory of ISC DHCP. In this file, we will add our local site configuration information to override the default settings in Makefile.conf.

- Edit the **site.conf** file (vi site.conf) and add the following parameters:

```
VARDB=/var/lib/dhcp
ADMMANDIR=/usr/share/man/man8
FFMANDIR=/usr/share/man/man5
LIBMANDIR=/usr/share/man/man3
USRMANDIR=/usr/share/man/man1
LIBDIR=/usr/lib
INCDIR=/usr/include
```

The above parameters specify where we want to install the program. For manual pages (man), we define the path to be under the /usr/share/man directory, for the DHCP database, we chose the /var/lib/dhcp directory, for the library files we locate everything under our lib directory (/usr/lib) and finally the include directory to be located under /usr/include.

These modifications are required to keep our path environment variables and filesystem definitions, under OpenNA Linux and Red Hat Linux, the same. If your Linux system is different, you may have to adjust the above definitions to fit your environment.

Step 5
The second source file to modify is called **site.h** and one of its functions is to specify the location of the dhcpd.leases and dhclient.leases files, which are used to store lease definitions of DHCP client connections. We'll change the default location for these files to be compliant with our Linux operating system again.

- Edit the **site.h** file (vi +108 includes/site.h) and change the line:

```
/* #define _PATH_DHCPD_DB       "/etc/dhcpd.leases" */
```

To read:

```
#define _PATH_DHCPD_DB       "/var/lib/dhcp/dhcpd.leases"
#define _PATH_DHCLIENT_DB    "/var/lib/dhcp/dhclient.leases"
```

Step 6
The hash table size feature used with DHCP plays an important role in respect to the performance of the DHCP server. The number of leases you expect to be assigned by the server is influenced by the size of the hash table, which can be customized in the includes/omapip/hash.h source file at compile time.

The default value assigned to this size is 9973 but depending of the number of clients that you expect to serve, the default value may be either too high or too low resulting in either an extremely large size that could take up more memory than may be necessary, or too small a size that could lead to a decrease in performance. The ideal situation would be to have the size of the hash table to be close to the number of leases you plan to have.

- Edit the **hash.h** file (vi +44 includes/omapip/hash.h) and change the line:

```
#define DEFAULT_HASH_SIZE      9973
```

To read:

```
#define DEFAULT_HASH_SIZE      200
```

In the above modification, we assume that you expect to serve between 1 and 200 client's with DHCP. If the number of clients' that you expect to serve is much higher, like 23000 for example, then change the value above to reflect this. The above hack, will have the effect to make the server perform faster, since it will spend less time traversing hash tables to try and find the lease it is looking for.

Step 7
Once the modifications have been made to the ISC DHCP source files, it is time to compile and optimize it for our system.

- To compile and optimize ISC DHCP use the following commands:
 ./configure --copts "-O2 -march=i686 -funroll-loops"

Step 8
At this stage the program is ready to be built and installed. We build ISC DHCP with the 'make' command and produce a list of files on the system before we install the software, and one afterwards, then compare them using the **diff** utility to find out what files were placed where and finally install ISC DHCP.

```
[root@deep dhcp-3.0p1]# make
[root@deep dhcp-3.0p1]# cd
[root@deep root]# find /* > DHCP1
[root@deep root]# cd /var/tmp/dhcp-3.0p1/
[root@deep dhcp-3.0p1]# make install
[root@deep dhcp-3.0p1]# strip /usr/sbin/dhcpd
[root@deep dhcp-3.0p1]# strip /usr/sbin/dhcrelay
[root@deep dhcp-3.0p1]# strip /sbin/dhclient
[root@deep dhcp-3.0p1]# touch /var/lib/dhcp/dhcpd.leases
[root@deep dhcp-3.0p1]# cd
[root@deep root]# find /* > DHCP2
[root@deep root]# diff DHCP1 DHCP2 > ISC-DHCP-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

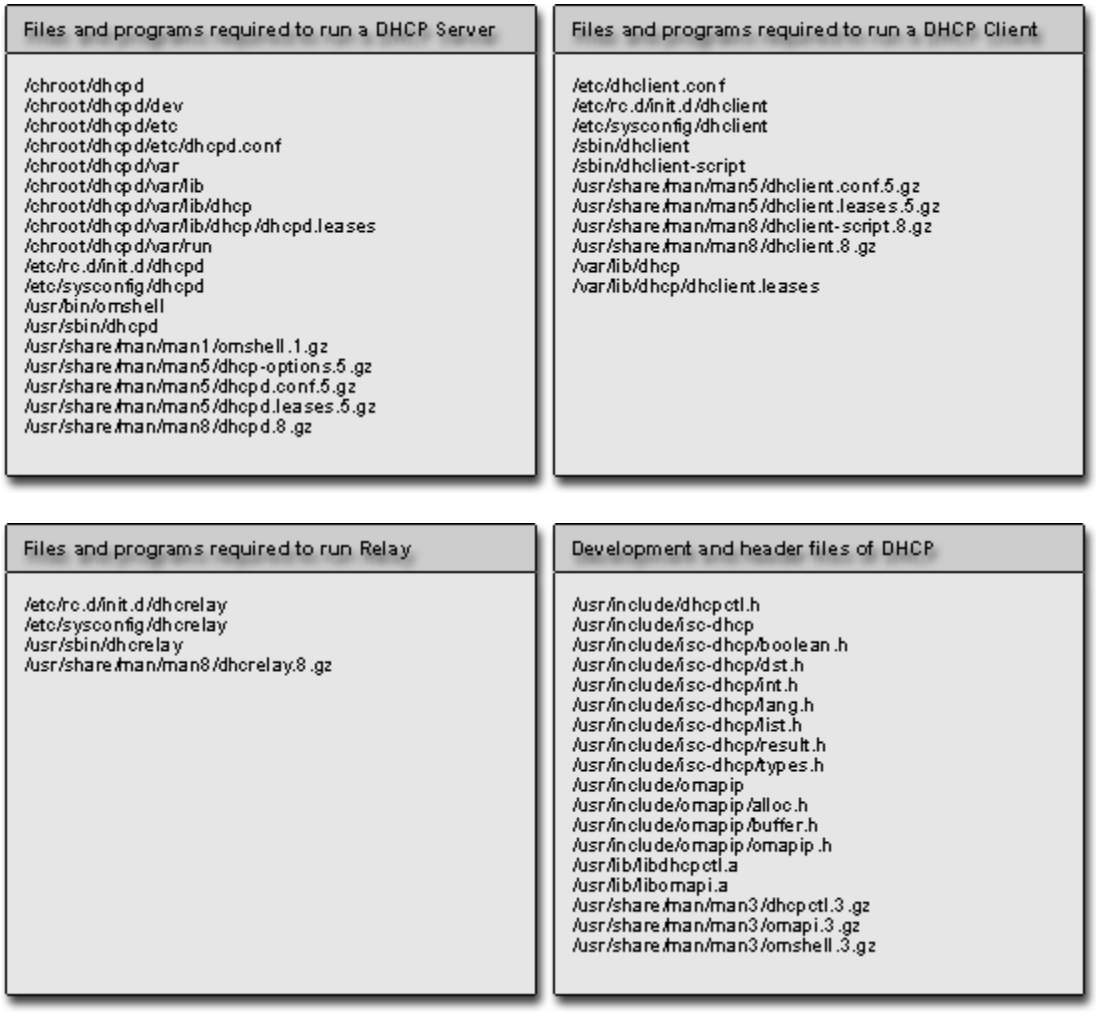
Step 9
Once the compilation, optimization and installation of the software has completed, we can free up some disk space by deleting the program tar archive and the related source directory, since they are no longer needed.

- To delete ISC BIND & DNS and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf dhcp-version/
[root@deep tmp]# rm -f dhcp-version.tar.gz
```

Step 10
Recall that DHCP can be installed to run as a DHCP Server, DHCP Client or with Relay agent. If you run DHCP as a server on your system, you don't need to keep the programs and files relating to the DHCP client, since your DHCP software will be configured to run as a server. The same is true for the DHCP client, if you want to run DHCP as a client for your system, you don't need to keep all the programs and files relating to the DHCP server. Finally, when you install a program from source code, all files related to this program are installed to the specified location on your system.

This means that even the headers and development files required to develop and link DHCP with other software are installed. This is good if you are developer and install DHCP on a development system, but if you just want to install DHCP for a specific utilization on a production server, you really don't need to keep the headers or development files, related to DCHP, installed on your computer. In the tables below, we show which files and programs are related to the DHCP server, client, relay agent and development. You can safely remove any files related to the DHCP that you don't run and only keep installed on your system the files and programs related to DHCP that you need.



Configuring ISC DHCP

After ISC DHCP has been built and installed successfully on your system, the next step is to configure and customize its configuration files to fit your needs.

- ✓ /etc/dhcpd.conf: (The ISC DHCP Configuration File)
- ✓ /etc/sysconfig/dhcpd: (The ISC DHCP System Configuration File)
- ✓ /etc/init.d/dhcpd: (The ISC DHCP Initialization File)

/etc/dhcpd.conf: The ISC DHCP Configuration File

The /etc/dhcpd.conf file is the main configuration file for ISC DHCP. It is in this configuration file that ISC DHCP gets all of its network information, the name of the server, the domain for which it is responsible, and so forth. Due to the nature of DHCP, we can be sure that we will have to configure network definitions such as subnet address, netmask, and a range of IP addresses.

The ISC DHCP configuration file has several customizable options and declarations available depending of the type of DHCP service that you want to offer. Here are the most important parameters to configure your DHCP server for maximum security; a complete listing and/or special requirements are available in the manual page for dhcpd.conf (5). We must configure the most important ones to suit our requirements and operating system. Text in bold are the parts of the configuration file that must be adjusted to suit your needs. Finally, we assume that you will run the DHCP software server on a Gateway Server with two network interfaces.

There are myriad of declarations, options and parameter available with DHCP. These may or may not be required depending of the type of DHCP server that you want for your network. In the configuration below, we cover most of the important parameters for a secure DHCP server that can be easily adjusted to fit a more complex network environment if required.

The goal for a secure configuration file with DHCP is to limit the complexity of the file to avoid errors induced by poor design and implementation. A typical dhcpd.conf file beginning with global parameters where we specify all definitions common for the entire configuration file and then continue with a list of statements, which fall into two broad categories - parameters and declarations.

- Create the **dhcpd.conf** file (touch /etc/dhcpd.conf). Below is what we recommend you set:

```
authoritative;
ddns-update-style none;
default-lease-time 400000;
max-lease-time 500000;
get-lease-hostnames true;
ping-check true;
deny bootp;

   if substring (option dhcp-client-identifier, 0, 4) = "RAS" {
        deny booting;
        }

   subnet 207.35.78.0 netmask 255.255.255.224 {
   not authoritative;
   }

   subnet 192.168.1.0 netmask 255.255.255.0 {
   option routers 192.168.1.254;

   pool {
   option domain-name "openna.com";
   option domain-name-servers 207.35.78.5, 207.35.78.6;
   option time-offset -18000;
   range 192.168.1.1 192.168.1.100;
 }
   }
```

This tells the dhcpd.conf file to set itself up for this particular configuration with:

authoritative;
The "authoritative;" statement is important and should be present at the top of your configuration file. It makes the DHCP server authoritative for all subnets. This means that if a client tries to renew an address that the server thinks is inappropriate, a DHCPNAK signal will be sent to the client machine. If this statement is not defined in the configuration file, clients will be unable to get a correct IP address after changing subnets until their old lease has expired. Therefore, it is important to have this statement defined at the top of your DHCP configuration file.

ddns-update-style none;
A DHCP server is able to dynamically assign domain name server information (DNS) to a client machine. This means that it can communicate with DNS servers to get the domain name server information and send them to the clients or get the information from its configuration file. The "ddns-update-style" statement specifies when we should allow a DNS update. For security reasons, we should NEVER allow DNS updates by the DHCP server. The parameter can be one of "ad-hoc", "interim" or "none". In our case we use "none", which means to deny DNS updates for clients on this DHCP server and inform DHCP to get its domain name information directly from its configuration file (the one we're configuring now).

default-lease-time 432000;
In DHCP terminology, clients "lease" IP addresses for a certain period of time. The "default-lease-time" statement specifies the default lease-time allowed for clients. After the time has elapsed the client will request a renewed lease from the server. Time is calculated in seconds, therefore, if we want to allow a lease-time of 5 days to clients, we will use the value of "432000" (60*60*24*5=432000). It is important to note that when the "default-lease-time" statement appears in the beginning of the configuration file or outside a subnet scope definition, it applies for all subnets defined in the configuration file unless overridden in a subnet scope.

max-lease-time 604800;
With ISC DHPC, clients can request leases of a specific duration, but to prevent machines from holding onto the lease forever, you can configure a maximum allowable lease time on the DHCP server. This is possible with the "max-lease-time" statement that specifies the maximum lease-time that clients can ask for when requesting more lease-time. As with the previous statement, time is calculated in seconds and, in our example, we allow a maximum lease-time of 7 days. Again, it is important to note that when the "max-lease-time" statement appears in the beginning of the configuration file or outside a subnet scope definition, it applies to all the subnet definitions in the configuration file, unless overridden in a subnet scope.

get-lease-hostnames true;
The "get-lease-hostnames" statement specifies if clients will be sent the DNS name associated with their allocated IP address or not. It is always a good idea to allow the server to send this information to clients and we answer "true" to this statement to allow it. The DNS information that it will be sending, are those that are specified later in this configuration file.

ping-check true;
The "ping-check" statement specifies if the server should check whether an address is in use before allocating one. If the value of this statement is "true", then the DHCP server will send an ICMP Echo request (a ping) to the address being assigned and then wait for a second to see if an ICMP Echo response has been received before assigning the address to the client.

deny bootp;
The "deny bootp" statement specifies to NOT support the bootp protocol unless overridden in a subnet scope. The bootp protocol is an old and dangerous protocol that should not be used on your network whenever possible. Note that this statement doesn't deny dynamic bootp clients in subnet pools if you are using the failover feature of DHCP. This is a security feature.

```
if substring (option dhcp-client-identifier, 0, 4) = "RAS" {
     deny booting;
     }
```

Microsoft Windowz NT comes pre-configured to allow 10 dial-in ports, which can quickly consume all the available DHCP addresses on the subnet. This is due to how the system manages the "**R**emote **A**ccess **S**erver" feature on Windows NT. To stop Windows NT RAS servers from consuming all the IP addresses, we define the above lines in the DHCP configuration file.

```
subnet 207.35.78.0 netmask 255.255.255.224 {
not authoritative;
}
```

The "subnet" declaration is one of the most important with DHCP and without it DHCP will not work. It is with this important declaration that we define to the DHCP server our network configuration. We use it to define the network and netmask IP address information that applies to the subnet that we want to provide the DHCP service. It is important to note that there are many options we can define in a subnet declaration. A subnet options declaration begins with an open bracket ({) and finishes with a close bracket (}). Being able to define options under a subnet declaration is useful when we have more than one subnet with different parameters but it's also useful even if we only have one subnet since it make the DHCP configuration file more readable.

In this example, we assume that you are installing the DHCP software on a Gateway Server (highly recommended) with two networks interfaces. One network interface for the external network, which gives us access to the Internet, and one for the internal network to access our private network. In the above statement, we define the subnet IP address of our external interface (subnet 207.35.78.0) with its corresponding netmask IP address (netmask 255.255.255.224) and set the external interface definition to NOT be authoritative and answer requests for the DHCP server (not authoritative;) for security reasons, since it is not wise to allow the external world to request dynamic IP addresses on our DHCP server. It is important to note that for completeness, we mention the first subnet on the external interface (eth0) even if we don't use it with DHCP. We define no address pools for this subnet, so no addresses can be allocated on this subnet.

```
subnet 192.168.1.0 netmask 255.255.255.0 {
```

The second "subnet" declaration define the network and netmask IP address information that applies to the subnet that we WANT to provide a DHCP service on. As you'll see, we will use some options, which apply to, and become specific for, this subnet declaration because they exist under the subnet options declaration.

We begin our subnet declaration definition by specifying the network (192.168.1.0) on which the DHCP server should answer and serve requests for dynamic IP addresses. We continue by defining the netmask address associated with the network (255.255.255.0). Please note that if you specify the wrong netmask address, DHCP will not work and will return an error message to your terminal and log file.

```
option routers 192.168.1.254;
```

Here another important parameter that should always exist on a subnet declaration for the DHCP server to work. This is our first option specific to our subnet declaration. The "option routers" specifies a list of IP addresses for routers on the client's subnet. Routers should be listed in order of preference. This means that you can have more than one router IP address defined here. If you have installed DHCP on your Gateway Server which has at least two network interfaces, then the router IP address should be the IP address of your internal interface (eth1) on the Gateway Server, which is connected to your internal private network.

```
pool {
```
We continue our explanation with another declaration in the subnet declaration. This is the "pool" declaration that can be used to specify a pool of addresses that will be treated differently than another pool of addresses, even on the same network segment or subnet. This means that we can define different options, range statements, etc into the same subnet declaration that applies to specific clients on the same subnet.

When all the options and parameters defined in the subnet declaration applies to the same client machines, it is not a requirement to use the "pool" declaration but a good idea if we want to be familiar with its use. A "pool" declaration begins with an open bracket ({) and finishes with a close bracket (}).

```
option domain-name "openna.com";
```
The above option is used in the "pool" declaration to make it specific to the client machines of this subnet declaration. The "domain-name" option is used to instruct the DHCP server that all clients should get this domain name by default, unless overridden in some other part of the dhcpd.conf file.

```
option domain-name-servers 207.35.78.5, 207.35.78.6;
```
The "domain-name-servers" option specifies which domain name servers all clients should get by default, unless overridden in some other part of the dhcpd.conf file. This allows the DHCP server to automatically assign domain name servers for the client's machine in the same way that it assigns IP addresses and network information.

When we define a domain name server with the above option, the DNS update feature should not be enabled. The DHCP server will get the DNS servers name from the above definition and send it to the client, who will then update the DNS server if configured to do so.

```
option time-offset -18000;
```
The "time-offset" option specifies the offset of the client's subnet in seconds from **C**oordinated **U**niversal **T**ime (UTC). Why would we need it? Remember that all times in the "dhcpd.leases" file are stored in the UTC (GMT) format because there is no portable way of storing leases in the local time zone. In the above definition, we set the value to -5 hours (60*60*5=18000) to reflect our time zone here in Quebec, Canada (GMT -5) and solve this problem.

```
range 192.168.1.1 192.168.1.100;
```
The "range" declaration specifies the range of IP addresses that the DHCP server can use to assign dynamic IP addresses to clients. We specify the values as the first available IP address to assign to clients and the last available IP address to assign to clients. Please note that in our example, not all of the addresses in that range are available, since the lowest address is 192.168.1.1 and the highest address is 192.168.1.100. This means that we allow the DHCP server to assign IP addresses starting from 192.168.1.1 to 192.168.1.100.

/etc/sysconfig/dhcpd: The DHCP System Configuration File

The /etc/sysconfig/dhcpd file is used to specify ISC DHCP system configuration information, such as if ISC DHCP should run in a chroot environment, print the entire DHCP copyright message on startup or if any other additional options are required to be passed to dhcpd daemon at startup.

- Create the **dhcpd** file (touch /etc/sysconfig/dhcpd) and add the following lines:

```
# Uncomment the following line to avoid printing the entire DHCP
# copyright message on startup.
#
#DHCPDARGS="-q"
```

The "DHCPARGS="-q"" option if uncommented, instructs ISC DHCP to avoid printing the entire DHCP copyright message on startup.

/etc/init.d/dhcpd: The DHCP Initialization File

The /etc/init.d/dhcpd script file is responsible for automatically starting and stopping the DHCP server. Loading the dhcpd daemon as a standalone daemon will eliminate load time and will even reduce swapping since non-library code will be shared. Please note that the following script is only suitable for Linux operating systems that use SystemV. If your Linux system uses some other method, like BSD, you'll have to adjust the script below to make it work for you.

Step 1
Create the **dhcpd** script file (touch /etc/init.d/dhcpd) and add the following lines:

```
#!/bin/bash

# This shell script takes care of starting and stopping DHCPD Server.
#
# chkconfig: 345 65 35
# description: Dhcpd provide access to Dynamic Host Control Protocol.
#
# processname: dhcpd

# Source function library.
. /etc/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Source for additional options if we have them.
if [ -f /etc/sysconfig/dhcpd ] ; then
        . /etc/sysconfig/dhcpd
fi

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

# If DHCPD is not available stop now.
[ -f /usr/sbin/dhcpd ] || exit 0
[ -f "${ROOTDIR}"/etc/dhcpd.conf ] || exit 0
[ -f "${ROOTDIR}"/var/lib/dhcp/dhcpd.leases ] || exit 0

# Path to the DHCPD binary.
dhcpd=/usr/sbin/dhcpd
```

```
RETVAL=0
prog="DHCPD"

start() {
        echo -n $"Starting $prog: "
        if [ -n "${ROOTDIR}" -a "x${ROOTDIR}" != "x/" ]; then
                DHCPDARGS="${DHCPDARGS} -chroot ${ROOTDIR}"
        fi
        daemon $dhcpd -user dhcpd -group dhcpd ${DHCPDARGS}
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/dhcpd
        return $RETVAL
}

stop() {
        echo -n $"Shutting down $prog: "
        killproc $dhcpd
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/dhcpd
        return $RETVAL
}

# See how we were called.
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  status)
        status $dhcpd
        RETVAL=$?
        ;;
  restart)
        stop
        start
        RETVAL=$?
        ;;
  condrestart)
        if [ -f /var/lock/subsys/dhcpd ]; then
                stop
                start
                RETVAL=$?
        fi
        ;;
  *)
        echo $"Usage: $0 {start|stop|status|restart|condrestart}"
        exit 1
esac
exit $RETVAL
```

Step 2
Once the dhcpd script file has been created, it is important to make it executable, change its default permissions, create the necessary links and then start it. Making this file executable will allow the system to run it, changing its default permission to allow only the root user to change it for security reasons, and the creation of the symbolic links will let the process control initialization of Linux, which is in charge of starting all the normal and authorized processes that need to run at boot time on your system, start the program automatically for you at each system reboot.

- To make this script executable and to change its default permissions, use the commands:
```
[root@deep /]# chmod 700 /etc/init.d/dhcpd
[root@deep /]# chown 0.0 /etc/init.d/dhcpd
```

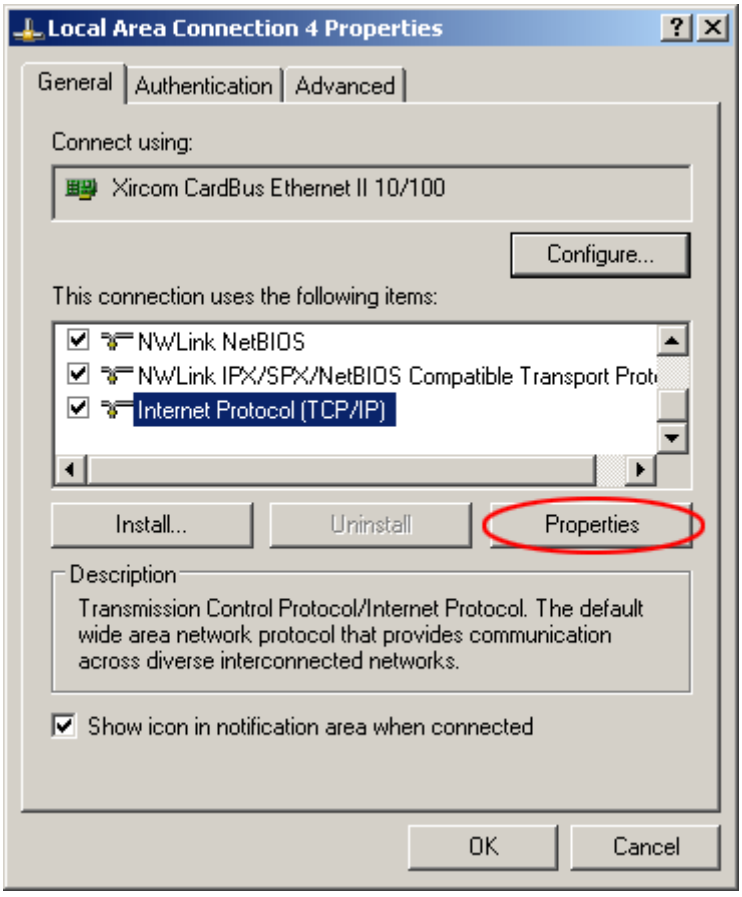
- To create the symbolic rc.d links for DHCP, use the following commands:
```
[root@deep /]# chkconfig --add dhcpd
[root@deep /]# chkconfig --level 345 dhcpd on
```

- To start DHCP software manually, use the following command:
```
[root@deep /]# /etc/init.d/dhcpd start
Starting Dhcpd:           [OK]
```

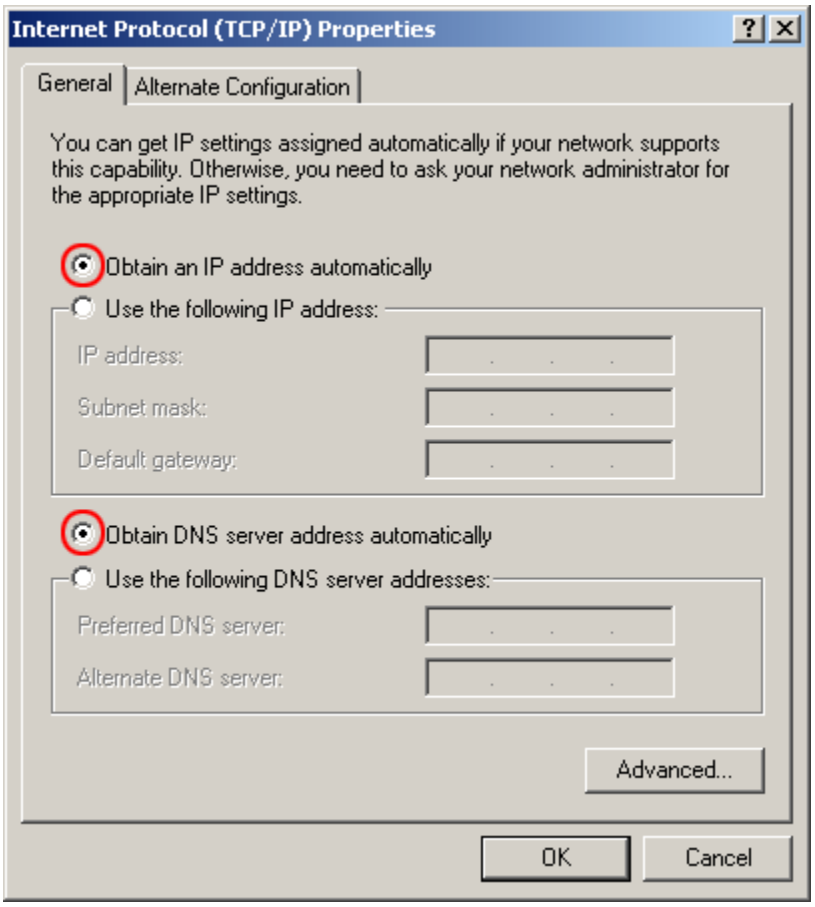
Testing the DHCP server

Once your DHCP server has been started, it's time to test it with some client machines to be sure that everything is running as expected. In our test, we will use a Windowz XP Pro system as a client to connect to the Internet.

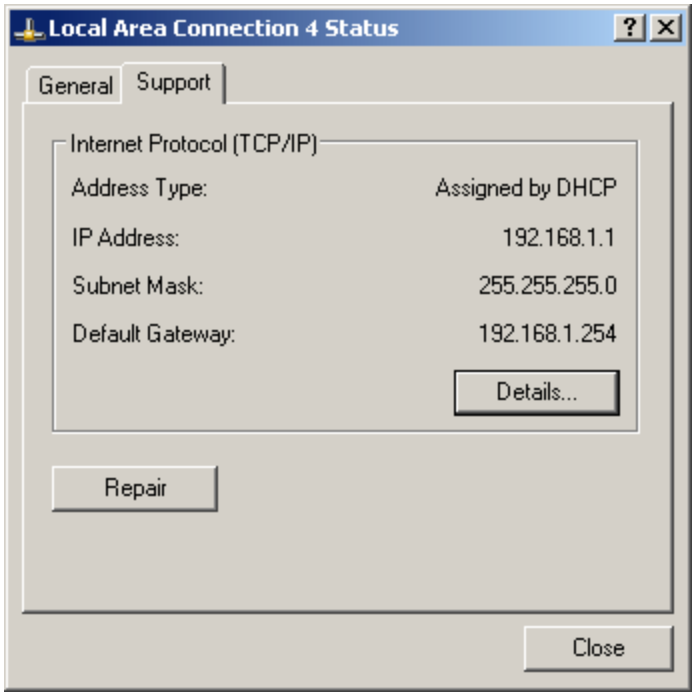
Step1
The first step to do is to open the window box relating to your network connection setup and click on the '**properties**' button to configure it.



Step2
Once the properties window is open, make sure that the radios buttons labeled '**Obtain an IP address automatically**' and '**Obtain DNS server address automatically**' are checked, then click on the '**OK**' button to close the window.



Step3
At this stage, you should have to reboot the system for the changes to take effect. Do it even if Windows doesn't do it for you. After all, it has been built for that. Once your Windows system has been rebooted check the status of your connection and if you see something like the following, then everything is ok and you should be able to browse the Internet.

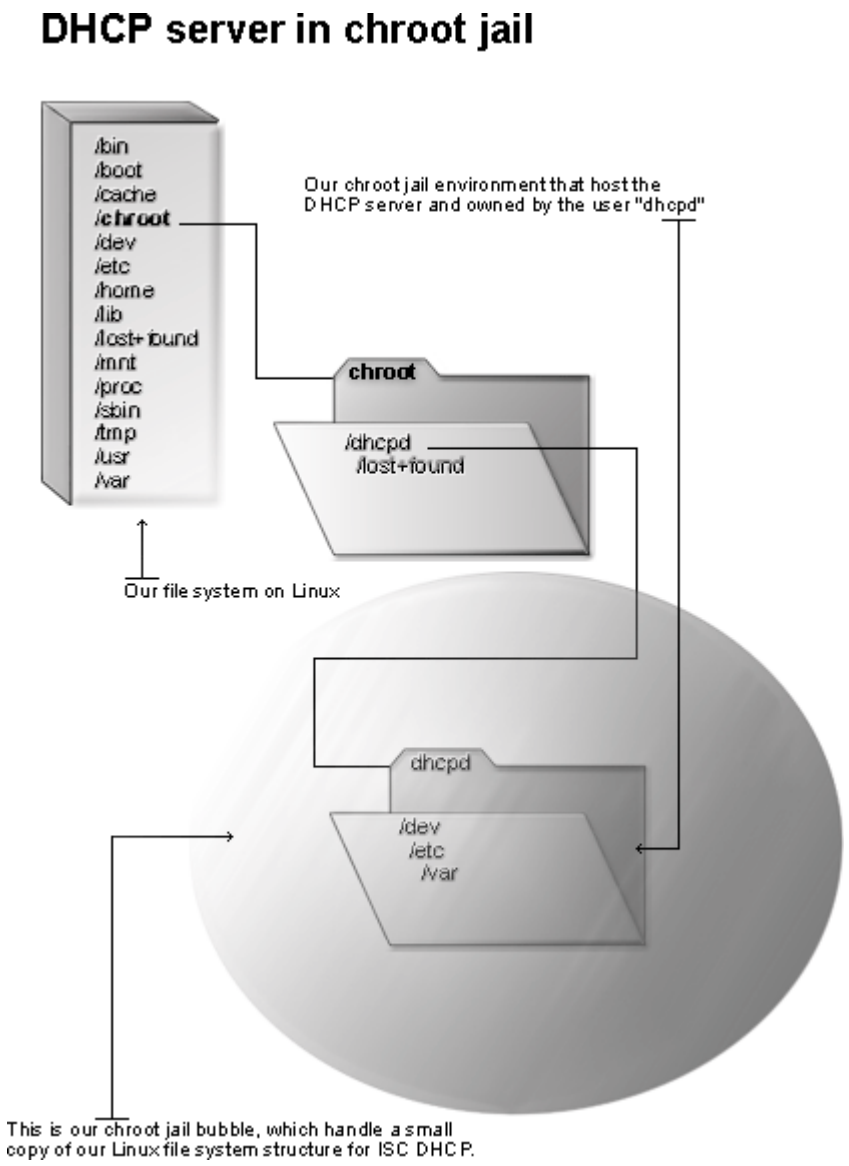


Running ISC DHCP in a chroot jail

This part focuses on preventing ISC DHCP from being used as a point of break-in to the system hosting it. For people who are paranoid about the installation of DHCP, it is possible to make it work in chroot jail environment. Contrary to some other UNIX software that only needs to be instructed to run in chroot mode, ISC DHCP needs to be patched before compilation, to run in this jail mode.

As we should know now, the main benefit of a chroot jail is that the jail will limit the portion of the file system the DHCP daemon program can see to the root directory of the jail. Additionally, since the jail only needs to support DHCP, the programs related to ISC DHCP available in the jail can be extremely limited. More importantly, there is no need for setuid-root programs, which can be used to gain root access and break out of the jail.

If the DHCP server is configured to run in chroot environment and a cracker breaks into the DHCP server, they will only be able to access and see the chroot environment where the DHCP server lives, and not the entire environment where the Linux operating system lives, reducing the possibility for the cracker to damage the system or to get 'root' access.


DHCP server in chroot jail
/bin
/boot
/cache
/chroot
/dev
/etc
/home
/lib
/lost+found
/mnt
/proc
/sbin
/tmp
/usr
/var
Our chroot jail environment that host the DHCP server and owned by the user "dhcpd"
chroot
/dhcpd
/lost+found
Our file system on Linux
dhcpd
/dev
/etc
/var
This is our chroot jail bubble, which handle a small copy of our Linux file system structure for ISC DHCP.

Necessary steps to run ISC DHCP software in a chroot jail:

The steps to run DHCP into a chroot environment differ in some ways from what we are accustomed to with other UNIX software. The reason for this difference is the fact that ISC DHCP is not built natively to run in this secure chroot mode. We have to patch the original source code of DHCP to make it possible. Once the program as been patched, compiled and installed, we need to create a skeleton root file system with enough components necessary (directories, files, etc.) to allow UNIX to do a chroot when the ISC DHCP daemon starts.

To run ISC DHCP in a chroot jail, we need to patch its source code and recompile the software. The procedure to compile and install the software is the same as explained, the difference being to patch the software and create some additional directories relating to the chroot environment. Again, we show you the steps from the beginning to avoid any confusion.

Step 1
The first step is to copy the software archive file to the /var/tmp directory and change to this location before expanding the archive.

- These procedures can be accomplished with the following commands:
```
[root@deep /]# cp dhcp-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf dhcp-version.tar.gz
```

Step 2
Once the archive has been expanded, we need to patch the source code of DHCP. Below is how to do it. Note: this file can also be downloaded from
http://www.episec.com/people/edelkind/patches/dhcp/dhcp-3.0+paranoia.patch.

- Create the **chroot.patch** file under the /var/tmp directory and move to this directory:
```
[root@deep /]# touch /var/tmp/chroot.patch
[root@deep /]# cd /var/tmp
```

- Edit the **chroot.patch** file you've just created (vi chroot.patch) and add the following lines:

```
--- dhcp-3.0/server/dhcpd.c     Thu Jun 21 22:12:58 2001
+++ dhcp-3.0+paranoia/server/dhcpd.c    Wed Oct 17 08:23:00 2001
@@ -56,6 +56,16 @@
 #include "version.h"
 #include <omapip/omapip_p.h>

+#if defined (PARANOIA)
+#  include <sys/types.h>
+#  include <unistd.h>
+#  include <pwd.h>
+/* get around the ISC declaration of group */
+#  define group real_group
+#    include <grp.h>
+#  undef group
+#endif /* PARANOIA */
+
 static void usage PROTO ((void));

 TIME cur_time;
@@ -204,6 +214,22 @@
        omapi_object_dereference (&listener, MDL);
 }
```

```
+#if defined (PARANOIA)
+/* to be used in one of two possible scenarios */
+static void setup_chroot (char *chroot_dir) {
+       if (geteuid())
+               log_fatal ("you must be root to use chroot");
+
+       if (chroot(chroot_dir)) {
+               log_fatal ("chroot(\"%s\"): %m", chroot_dir);
+       }
+       if (chdir ("/")) {
+               /* probably permission denied */
+               log_fatal ("chdir(\"/\"): %m");
+       }
+}
+#endif /* PARANOIA */
+
 int main (argc, argv, envp)
        int argc;
        char **argv, **envp;
@@ -236,6 +262,14 @@
        char *traceinfile = (char *)0;
        char *traceoutfile = (char *)0;
 #endif
+#if defined (PARANOIA)
+       char *set_user   = 0;
+       char *set_group  = 0;
+       char *set_chroot = 0;
+
+       uid_t set_uid = 0;
+       gid_t set_gid = 0;
+#endif /* PARANOIA */

        /* Make sure we have stdin, stdout and stderr. */
        status = open ("/dev/null", O_RDWR);
@@ -298,6 +332,20 @@
                        if (++i == argc)
                                usage ();
                        server = argv [i];
+#if defined (PARANOIA)
+               } else if (!strcmp (argv [i], "-user")) {
+                       if (++i == argc)
+                               usage ();
+                       set_user = argv [i];
+               } else if (!strcmp (argv [i], "-group")) {
+                       if (++i == argc)
+                               usage ();
+                       set_group = argv [i];
+               } else if (!strcmp (argv [i], "-chroot")) {
+                       if (++i == argc)
+                               usage ();
+                       set_chroot = argv [i];
+#endif /* PARANOIA */
                } else if (!strcmp (argv [i], "-cf")) {
                        if (++i == argc)
                                usage ();
@@ -397,6 +445,44 @@
                                                trace_seed_stop, MDL);
 #endif

+#if defined (PARANOIA)
+       /* get user and group info if those options were given */
+       if (set_user) {
```

```
+               struct passwd *tmp_pwd;
+
+               if (geteuid())
+                       log_fatal ("you must be root to set user");
+
+               if (!(tmp_pwd = getpwnam(set_user)))
+                       log_fatal ("no such user: %s", set_user);
+
+               set_uid = tmp_pwd->pw_uid;
+
+               /* use the user's group as the default gid */
+               if (!set_group)
+                       set_gid = tmp_pwd->pw_gid;
+       }
+
+       if (set_group) {
+/* get around the ISC declaration of group */
+#define group real_group
+               struct group *tmp_grp;
+
+               if (geteuid())
+                       log_fatal ("you must be root to set group");
+
+               if (!(tmp_grp = getgrnam(set_group)))
+                       log_fatal ("no such group: %s", set_group);
+
+               set_gid = tmp_grp->gr_gid;
+#undef group
+       }
+
+#  if defined (EARLY_CHROOT)
+       if (set_chroot) setup_chroot (set_chroot);
+#  endif /* EARLY_CHROOT */
+#endif /* PARANOIA */
+
        /* Default to the DHCP/BOOTP port. */
        if (!local_port)
        {
@@ -500,6 +586,10 @@

        postconf_initialization (quiet);

+#if defined (PARANOIA) && !defined (EARLY_CHROOT)
+       if (set_chroot) setup_chroot (set_chroot);
+#endif /* PARANOIA && !EARLY_CHROOT */
+
         /* test option should cause an early exit */
        if (cftest && !lftest)
                exit(0);
@@ -543,6 +633,22 @@
                        exit (0);
        }

+#if defined (PARANOIA)
+       /* change uid to the specified one */
+
+       if (set_gid) {
+               if (setgroups (0, (void *)0))
+                       log_fatal ("setgroups: %m");
+               if (setgid (set_gid))
+                       log_fatal ("setgid(%d): %m", (int) set_gid);
+       }
+
```

```
+       if (set_uid) {
+               if (setuid (set_uid))
+                       log_fatal ("setuid(%d): %m", (int) set_uid);
+       }
+#endif /* PARANOIA */
+
        /* Read previous pid file. */
        if ((i = open (path_dhcpd_pid, O_RDONLY)) >= 0) {
                status = read (i, pbuf, (sizeof pbuf) - 1);
@@ -888,6 +994,10 @@

        log_fatal ("Usage: dhcpd [-p <UDP port #>] [-d] [-f]%s%s%s%s",
                   "\n             [-cf config-file] [-lf lease-file]",
+#if defined (PARANOIA)
+                  /* meld into the following string */
+                  "\n             [-user user] [-group group] [-chroot
dir]"
+#endif /* PARANOIA */
 #if defined (TRACING)
                   "\n             [-tf trace-output-file]",
                   "\n             [-play trace-input-file]"
```

- Move into the `dhcp-3.0p1` source directory and patch your source code:

```
[root@deep tmp]# cd /var/tmp/dhcp-3.0p1/
[root@deep dhcp-3.0p1]# patch -p1 < ../chroot.patch
```

NOTE: The above patch or any update can be retrieved at the following URL:
`http://www.episec.com/people/edelkind/patches/dhcp/dhcp-3.0+paranoia.patch`.

Step 3
Now perform the following steps before compiling and optimizing `DHCP`. To recap, these are the same modifications as shown at the beginning of this chapter. If you need more information about their meaning, please see the instructions earlier in this chapter.

We must modify the **`site.conf`** file located under the source directory of `ISC DHCP`. In this file, we will add our local site configuration settings to override the default ones in `Makefile.conf`.

- Edit the **`site.conf`** file (`vi site.conf`) and add the following parameters:

```
VARDB=/var/lib/dhcp
ADMMANDIR=/usr/share/man/man8
FFMANDIR=/usr/share/man/man5
LIBMANDIR=/usr/share/man/man3
USRMANDIR=/usr/share/man/man1
LIBDIR=/usr/lib
INCDIR=/usr/include
```

Step 4
Another source file to modify is **site.h** and one of its functions is to specify the location of the dhcpd.leases and dhclient.leases files, which are used to store the lease definitions for DHCP client connections. We'll change the default location for these files to be compliant with our Linux system environment.

- Edit the **site.h** file (vi +108 includes/site.h) and change the line:

```
/* #define _PATH_DHCPD_DB       "/etc/dhcpd.leases" */
```

To read:

```
#define _PATH_DHCPD_DB       "/var/lib/dhcp/dhcpd.leases"
#define _PATH_DHCLIENT_DB    "/var/lib/dhcp/dhclient.leases"
```

Step 5
The hash table size feature used with DHCP plays an important part in the performance of the DHCP server. The number of leases you expect to assign on the server is influenced by the size of the hash table, which can be customized in the includes/omapip/hash.h source file at compile time.

The default value assigned to this size is 9973 but depending on the number of clients that you expect to serve, the default value may be either too high or too low resulting to an extremely large size that could takes up more memory than may be necessary, or to a small size that could decrease performance. The ideal situation would be to have the size of the hash table to be close to the number of leases you plan to have.

- Edit the **hash.h** file (vi +44 includes/omapip/hash.h) and change the line:

```
#define DEFAULT_HASH_SIZE      9973
```

To read:

```
#define DEFAULT_HASH_SIZE      200
```

With the above modification, we presume that you expect to serve between 1 and 200 clients with DHCP. If the number of client machines that you expect to serve is really higher like 23000 for example, then change the above value to reflect this number. The above hack, will have the effect of making the server perform faster, since it will spend less time traversing hash tables to try and find the lease it is looking for.

Step 6
Once the modifications have been made to the source files, we can compile and optimize it for our system. Here are some differences with the previous compilation lines that we have used previously in this chapter for the DHCP software without the chroot jail feature. We need to add to the "--copts" option two additional options to make DHCP to compile with chroot environment feature enable.

- To compile and optimize ISC DHCP with chroot jail support, use the following commands:

```
./configure --copts "-DEARLY_CHROOT -DPARANOIA -O2 -march=i686 -funroll-loops"
```

Step 7
Now, the program is ready to be built and installed. We build it with the 'make' command and produce a list of files on the system before we install the software, and one afterwards, then compare them using the **diff** utility to find out what files are placed where and finally install ISC DHCP.

```
[root@deep dhcp-3.0p1]# make
[root@deep dhcp-3.0p1]# cd
[root@deep root]# find /* > DHCP1
[root@deep root]# cd /var/tmp/dhcp-3.0p1/
[root@deep dhcp-3.0p1]# make install
[root@deep dhcp-3.0p1]# strip /usr/sbin/dhcpd
[root@deep dhcp-3.0p1]# strip /usr/sbin/dhcrelay
[root@deep dhcp-3.0p1]# strip /sbin/dhclient
[root@deep dhcp-3.0p1]# cd
[root@deep root]# find /* > DHCP2
[root@deep root]# diff DHCP1 DHCP2 > ISC-DHCP-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

Step 8
Once compilation, optimization and installation of the software has finished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete ISC BIND & DNS and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf dhcp-version/
[root@deep tmp]# rm -f dhcp-version.tar.gz
```

Step 9
At this stage, we need to setup the chroot environment, and create the root directory of the jail. We've chosen /chroot/dhcpd for this because we want to put this on its own separate file system to prevent file system attacks. Earlier, in our Linux installation procedure, we created a special partition /chroot for this purpose.

```
[root@deep /]# mkdir -p /chroot/dhcpd/etc
[root@deep /]# mkdir -p /chroot/dhcpd/dev
[root@deep /]# mkdir -p /chroot/dhcpd/var/run
[root@deep /]# mkdir -p /chroot/dhcpd/var/lib/dhcp
[root@deep /]# touch /chroot/dhcpd/var/lib/dhcp/dhcpd.leases
[root@deep /]# cd /chroot/dhcpd/var/
[root@deep var]# chown -R dhcpd.dhcpd /lib/dhcp/dhcpd.leases
[root@deep var]# chown -R dhcpd.dhcpd /run/
[root@deep var]# chown -R dhcpd.dhcpd /lib/
```

We need all of the above directories because, from the point of the chroot, we're sitting at "/" and anything above this directory is inaccessible.

Step 10
After that, we must move the configuration file of ISC DHCP and create the log character device in the appropriate places in the chroot jail.

```
[root@deep /]# mv /etc/dhcpd.conf /chroot/dhcpd/etc/
[root@deep /]# /dev/MAKEDEV -d /chroot/dhcpd/dev/ -m 1 log
[root@deep /]# /bin/chmod 0666 /chroot/dhcpd/dev/log
[root@deep /]# /bin/rm -f /chroot/dhcpd/dev/logibm
[root@deep /]# /bin/rm –rf /chroot/dhcpd/dev/logicalco/
[root@deep /]# /bin/rm –f /chroot/dhcpd/dev/logimouse
```

In the above commands, we move our DHCP configuration file to the /chroot/dhcpd/etc directory and create a character device called "log" under the /chroot/dhcpd/dev directory, change its mode permission then remove the "logibm", "logicalco", and "logimouse" character devices and directory, which was created by the 'MAKEDEV' command since we don't need them.

Step 11
For additional security, we can 'chattr' the dhcpd.conf file in the chroot jail directory.

- This procedure can be accomplished with the following commands:
```
[root@deep /]# cd /chroot/named/etc/
[root@deep etc]# chattr +i dhcpd.conf
```

WARNING: Don't forget to remove the immutable bit on this file if you have to make some modifications to it later, use the command "chattr –i".

Step 12
At this point, we have to instruct ISC DHCP to start in the chroot jail environment. This is done by modifying our original /etc/sysconfig/dhcpd and /etc/init.d/dhcpd script files. We start with our dhcpd file under the /etc/sysconfig directory and continue with our /etc/init.d/dhcpd initialization script file.

- Edit the **dhcpd** file (vi /etc/sysconfig/dhcpd) and add/change the following lines:

```
# This option will run dhcpd in a chroot environment.
#
ROOTDIR="/chroot/dhcpd"

# Uncomment the following line to avoid printing the entire DHCP
# copyright message on startup.
#
DHCPDARGS="-q"
```

The "ROOTDIR="/chroot/dhcpd/"" option instructs ISC DHCP where the chroot directory is located. Therefore the dhcpd daemon reads this line in the /etc/sysconfig/dhcpd file and chroot's to the specified directory before starting.

- Edit the **dhcpd** file (`vi /etc/init.d/dhcpd`) and add/change the following lines:

```
#!/bin/bash

# This shell script takes care of starting and stopping DHCPD Server.
#
# chkconfig: 345 65 35
# description: Dhcpd provide access to Dynamic Host Control Protocol.
#
# processname: dhcpd

# Source function library.
. /etc/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Source for additional options if we have them.
if [ -f /etc/sysconfig/dhcpd ] ; then
        . /etc/sysconfig/dhcpd
fi

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

# If DHCPD is not available stop now.
[ -f /usr/sbin/dhcpd ] || exit 0
[ -f "${ROOTDIR}"/etc/dhcpd.conf ] || exit 0
[ -f "${ROOTDIR}"/var/lib/dhcp/dhcpd.leases ] || exit 0

# Path to the DHCPD binary.
dhcpd=/usr/sbin/dhcpd

RETVAL=0
prog="DHCPD"

start() {
        echo -n $"Starting $prog: "
        if [ -n "${ROOTDIR}" -a "x${ROOTDIR}" != "x/" ]; then
                DHCPDARGS="${DHCPDARGS} -chroot ${ROOTDIR}"
        fi
        daemon $dhcpd -user dhcpd -group dhcpd ${DHCPDARGS}
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/dhcpd
        return $RETVAL
}

stop() {
        echo -n $"Shutting down $prog: "
        killproc $dhcpd
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/dhcpd
        return $RETVAL
}

# See how we were called.
case "$1" in
  start)
```

```
        start
        ;;
  stop)
        stop
        ;;
  status)
        status $dhcpd
        RETVAL=$?
        ;;
  restart)
        stop
        start
        RETVAL=$?
        ;;
  condrestart)
        if [ -f /var/lock/subsys/dhcpd ]; then
                stop
                start
                RETVAL=$?
        fi
        ;;
  *)
        echo $"Usage: $0 {start|stop|status|restart|condrestart}"
        exit 1
esac
exit $RETVAL
```

Step 13
Finally, we must test the new chrooted jail configuration of our ISC DHCP server.

- Start the new chrooted jail ISC DHCP software with the following command:
  ```
  [root@deep /]# /etc/init.d/dhcpd start
  Starting Dhcpd:                          [OK]
  ```

- If you don't get any errors, do a '**ps ax | grep dhcpd**' and see if we're running:
  ```
  [root@deep /]# ps ax | grep dhcpd
  9785 ?        S      0:00 /usr/sbin/dhcpd -user dhcpd -group dhcpd -
  chroot /chroot/dhcpd
  ```

If so, let's check to make sure it's chrooted by picking out its process numbers and doing '**ls -la /proc/that_process_number/root/**'.

```
[root@deep /]# ls -la /proc/9785/root/
```

If you see something like:

```
drwxr-xr-x    5 root     root         4096 Feb 22 04:56 ./
drwxr-xr-x    4 root     root         4096 Feb 22 04:56 ../
drwxr-xr-x    2 root     root         4096 Feb 22 04:56 dev/
drwxr-xr-x    2 root     root         4096 Feb 22 04:56 etc/
drwxr-xr-x    4 root     root         4096 Feb 22 04:56 var/
```

Congratulations! Your ISC DHCP server in chroot jail is working.

Securing ISC DHCP

This section deals specifically with actions we can take to improve security under ISC DHCP. The interesting points here are that we refer to the features available within the base installed program and not to any additional software.

Assigning a fixed-address to clients machine:

For additional security measures and on networks where there are not hundred of clients requesting dynamic IP addresses, the administrator can decide to use fixed-addresses to allow workstations to get an IP address from the DHCP server. This is possible by defining a host declaration for each client machine that need to get a valid dynamic assigned IP address. On a large corporate network, this solution is simply not viable and should be excluded.

Step1

The steps to achieve this are really simple. You have to declare the following lines in your DHCP configuration file for each additional workstation that need a dynamic IP address from the DHCP server. Without these lines, nothing will work and your clients' machine will not be able to get an IP address from the DHCP server.

We will use our original dhcpd.conf file as described earlier in this chapter and will add the lines below to declare a fixed-address to a client machine allowed to get a valid IP address. Text in bold is what you should add to your default DHCP configuration file to make it work.

- Edit your original **dhcpd.conf** file (vi /etc/dhcpd.conf) and add the following lines. Below is what we recommend you change. You will have to change the example parameters to reflect your own settings or it will not work.

```
authoritative;
ddns-update-style none;
default-lease-time 400000;
max-lease-time 500000;
get-lease-hostnames true;
ping-check true;
deny bootp;

   if substring (option dhcp-client-identifier, 0, 4) = "RAS" {
        deny booting;
        }

   subnet 207.35.78.0 netmask 255.255.255.224 {
   not authoritative;
   }

   subnet 192.168.1.0 netmask 255.255.255.0 {
   option routers 192.168.1.254;

   pool {
   option domain-name "openna.com";
   option domain-name-servers 207.35.78.5, 207.35.78.6;
   option time-offset -18000;
   range 192.168.1.1 192.168.1.100;

   host desk1 {
   hardware ethernet 02:30:b6:4a:31:9e;
   fixed-address 192.168.1.1;
   }

   deny unknown-clients;

 }
  }
```

`host desk1 {`
The "`host`" statement is used to define the hostname identifying the workstation that wants to get its `IP` address from the `DHCP` server. The hostname of this workstation should be a valid hostname. In general all workstations and servers should have at least a host name assigned by the administrator to distinguish them.

`hardware ethernet 02:30:b6:4a:31:9e;`
In order for a client machine to be recognized and allowed to request an `IP` address from the `DHCP` server, its network hardware address (`MAC`) must be declared using a `hardware` clause in the `host` statement. We declare the value by specifying the name of a physical hardware interface type, which can be either "`ethernet`" or "`token-ring`". Next, we have to define the hardware-address of the network interface in hexadecimal octets (`02:30:b6:4a:31:9e`). On Linux system, you can find the hexadecimal octets of the network interface with the "`ifconfig`" command, on Windows systems you have to check on the networking properties of the network card.

`fixed-address 192.168.1.1;`
The "`fixed-address`" statement is used to assign one or more fixed `IP` addresses to a client machine and should only appear in a host declaration. It is this fixed `IP` address the `DHCP` server will return to the workstation. Therefore, the value you specify here is what your workstation will get as `IP` address from the `DHCP` server.

`deny unknown-clients;`
The last parameter in our configuration file is the "`deny unknown-clients`" option. This option instructs the `DHCP` server to deny dynamically assigned `IP` addresses to all unknown clients. Remember that an unknown client is simply a client machine who doesn't have a host name declaration defined in the `DHCP` configuration file as shown above. Finally, it is important to note that the "`deny unknown-clients`" statement should appear ONLY once in the configuration even if you declare myriad of hosts. DON'T define this statement every time you add a new fixed-address for a workstation into your configuration file.

Step 2
Once you have added your new fixed-address for your client machine into the `DHCP` configuration file, you must restart the `DHCP` server for the changes to take effect.

- To restart the `DHCP` server, use the following command:

```
[root@deep /]# /etc/init.d/dhcpd restart
Stoping  Dhcpd:           [OK]
Starting Dhcpd:           [OK]
```

Running the `DHCP` client for Linux

When we've compiled `DHCP`, we have installed the whole package on our server. This means that the `DHCP` client software is available to be used on the `DHCP` server, but we really don't need it on the `DHCP` server as it's only required on Linux systems that don't act as `DHCP` servers. The reason is obvious, a `DHCP` server handles `DHCP` client requests and assigns dynamic `IP` address to the requesting client, therefore it doesn't need to have programs related to the `DHCP` client installed but only the programs required for a `DHCP` server to function.

This means that we have to reinstall DHCP on the Linux system where we want to use it as DHCP client and remove all files and binaries relating to DHCP running in server mode. A DHCP client should be used on Linux systems when we need to establish a connection with DHCP server to get a dynamically assigned IP address. This can arise because we want to get an IP address from our own DHCP server or from our ISP. The concept is the same; a DHCP client will make the connection to a DHCP server and request a dynamic IP address to access the network or the Internet.

Step1
In the steps below, we assume that you have reinstalled DHCP as explained earlier in this chapter and have removed the files and programs relating to the DHCP server and DHCP relay agent. Remember, you don't need to keep all of the installed DHCP files on a Linux client machine to make it work as DHCP client, but only the programs and files required for a DHCP client to run. Here are all the files, programs and manual pages that you should keep on your client machine to make the DHCP client work.

```
/etc/dhclient.conf
/var/lib/dhcp/dhclient.leases
/sbin/dhclient
/sbin/dhclient-script
/usr/share/man/man5/dhclient.conf.5.gz
/usr/share/man/man5/dhclient.leases.5.gz
/usr/share/man/man8/dhclient-script.8.gz
/usr/share/man/man8/dhclient.8.gz
```

As with ISC DHCP running in DHCP server mode, a DHCP client needs some configuration files to run. Once the required DHCP client software has been installed, your next step is to configure it's the configuration files to fit your needs. These configuration files are:

- ✓ /etc/dhclient.conf (The ISC DHCP Client Configuration File)
- ✓ /etc/sysconfig/dhclient (The ISC DHCP Client System Configuration File)
- ✓ /etc/init.d/dhclient (The ISC DHCP Client Initialization File)

/etc/dhclient.conf: The DHCP Client Configuration File

The /etc/dhclient.conf file is the configuration file for the ISC DHCP client. It is in this file that we configure the behavior of the client. The DHCP client configuration file is really easy to configure and in many cases, it's sufficient to use an empty dhclient.conf file to make the DHCP client to work. Yes, you can test it, just create an empty dhclient.conf file and start your DHCP client; you'll see that the client machine automatically gets all of its network information from the DHCP server without a problem.

Therefore why do we need to customize the dhclient.conf file if a DHCP client can work with an empty DHCP client configuration file? A DHCP client can automatically request all relevant network information from the DHCP server for most network requirements, but sometimes we can have a special DHCP network architecture where, for example, two DHCP servers live and assign dynamic IP addresses to clients on the same network. In this case, a DHCP client without a customized dhclient.conf file cannot differentiate between either DHCP server and will try to get network information from the first DHCP server who responds to its request. Another case is when you have two networks cards installed on the same system and want to inform the DHCP client software to use one Ethernet card in particular and forget the other. These are just two situations that I can think of, but others probably exist. If you are in this kind of situation a DHCP client configuration file should by used and you will have to consult the 'dhclient.conf' manual page on your system for more information on the required options and statements.

As stated earlier, in most cases an empty dhclient.conf file, as shown below, will be sufficient.

- Create the **dhclient.conf** file (touch /etc/dhclient.conf) and add the lines:

```
# This configuration file is empty because in many cases
# you don't need to configure anything for DHCP client
# to work on your system.
```

/etc/sysconfig/dhcliemt: The DHCP Client System Configuration File

The /etc/sysconfig/dhclient file is used to specify ISC DHCP client system configuration information. We use it to define additional options that can be passed to the dhclient daemon program at startup. In the dhclient system configuration file below, we define an option that allows us to print or suppress the entire DHCP copyright message on startup.

- Create the **dhclient** file (touch /etc/sysconfig/dhclient) and add the following lines:

```
# Uncomment the following line to avoid printing the entire DHCP
# copyright message on startup.
#
#DHCPDARGS="-q"
```

As with the other DHCP system configuration files, the "DHCPARGS="-q"" option if uncommented, instructs ISC DHCP to avoid printing the entire DHCP copyright message on startup. It is a good idea to enable this option if we want to have a clean log file report on DHCP.

/etc/init.d/dhclient: The DHCP Client Initialization File

The /etc/init.d/dhclient script file is responsible for automatically starting and stopping the DHCP client software on your Linux system. Please note that the following script is suitable for Linux operating systems that use SystemV. If you Linux system use another method like BSD, you'll have to adjust the script below to make it work for you.

Step 1
Create the **dhclient** script file (touch /etc/init.d/dhclient) and add the following lines:

```
#!/bin/bash

# This shell script takes care of starting and stopping DHCP Client.
#
# chkconfig: 345 65 35
# description: Dhclient provide client access to Dynamic Host Control Protocol.
#
# processname: dhclient

# Source function library.
. /etc/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Source for additional options if we have them.
if [ -f /etc/sysconfig/dhclient ] ; then
```

```
        . /etc/sysconfig/dhclient
fi

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

# If DHCP Client is not available stop now.
[ -f /sbin/dhclient ] || exit 0
[ -f /var/lib/dhcp/dhclient.leases ] || exit 0

# Path to the DHCP Client binary.
dhclient=/sbin/dhclient

RETVAL=0
prog="DHCP Client"

start() {
        echo -n $"Starting $prog: "
        daemon $dhclient ${DHCPDARGS}
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/dhclient
        return $RETVAL
}

stop() {
        echo -n $"Shutting down $prog: "
        killproc $dhclient
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/dhclient
        return $RETVAL
}

# See how we were called.
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  status)
        status $dhclient
        RETVAL=$?
        ;;
  restart)
        stop
        start
        RETVAL=$?
        ;;
  condrestart)
        if [ -f /var/lock/subsys/dhclient ]; then
                stop
                start
                RETVAL=$?
        fi
        ;;
  *)
        echo $"Usage: $0 {start|stop|status|restart|condrestart}"
        exit 1
esac
exit $RETVAL
```

Step 2
Once the dhclient script file has been created, it is important to make it executable, change its default permissions, create the necessary links and start it. Making this file executable will allow the system to run it, change its default permissions to allow only the root user to change this file for security reasons, and creation of the symbolic links will let the process control initialization of Linux which is in charge of starting all the normal and authorized processes that need to run at boot time on your system to start the program automatically for you at each system reboot.

- To make this script executable and to change its default permissions, use the commands:
```
[root@deep /]# chmod 700 /etc/init.d/dhclient
[root@deep /]# chown 0.0 /etc/init.d/dhclient
```

- To create the symbolic rc.d links for DHCP Client, use the following commands:
```
[root@deep /]# chkconfig --add dhclient
[root@deep /]# chkconfig --level 345 dhclient on
```

- To start the DHCP Client software manually, use the following command:
```
[root@deep /]# /etc/init.d/dhclient start
Starting Dhclient:           [OK]
```

NOTE: A RPM package called "pump" exists on the Linux CD-ROM. The "pump" package is a combined BOOTP and DHCP client daemon, which allows your machine to retrieve configuration information from a DHCP server. The difference with the DHCP client software from the ISC group is that "pump" is supposed to be smaller and faster. In any case both work fine. It is yours to decide which one is best for you.

Further documentation

For more details, there are some manual pages about DHCP that you could read:

```
$ man omshell (1)           - OMAPI Command Shell.
$ man dhcp-options (5)      - Dynamic Host Configuration Protocol options.
$ man dhcpd.conf (5)        - dhcpd configuration file.
$ man dhcpd.leases (5)      - DHCP client lease database.
$ man dhcpd (8)             - Dynamic Host Configuration Protocol Server.
```



Part IX

Mail Transfer Agent Protocol

CHAPTER 30

Exim

IN THIS CHAPTER

1. **Compiling - Optimizing & Installing `Exim`**
2. **Configuring `Exim`**
3. **Testing `Exim`**
4. **Allowing Users to authenticate with `Exim` before relaying**
5. **Running `Exim` with `SSL` support**
6. **Running `Exim` with Virtual Hosts support**
7. **Running `Exim` with Maildir support**
8. **Running `Exim` with mail quota support**
9. **Running `Exim` as a Null Client Mail Server**
10. **`Exim` Administrative Tools**

Linux Exim

Abstract

Wherein this chapter we'll talk about mail and the necessity of having a mail server installed on our secure Linux server. On all kinds of machines that run a UNIX operating system it's necessary and NOT optional to have a mail server. Even if you don't set-up your system to send or receive mail for users, you'll always have possible log messages that need to be delivered to root user, postmaster, daemons program, etc. This is where a mail server is vital or you may lose some important messages like errors, attacks, intrusions etc. The next two chapters will deal extensively with **M**ail **T**ransport **A**gents you may want to install. We will begin our reading with Exim and finish with Qmail software.

Exim is a **M**ail **T**ransfer **A**gent (MTA) developed at the University of Cambridge for use on UNIX systems connected to the Internet. It contains facilities for verifying incoming sender and recipient addresses, refusing mail from specified hosts, networks, or senders, and for controlling mail relaying. The purpose of an MTA is to send mail from one machine to another, and nothing else. Exim is not a client program, which you use to read your e-mail. Instead, it actually moves your email over networks, or the Internet, to where you want it to go. Exim is one of the most, if not the most, secure mail transfer agent available on the Internet for UNIX systems.

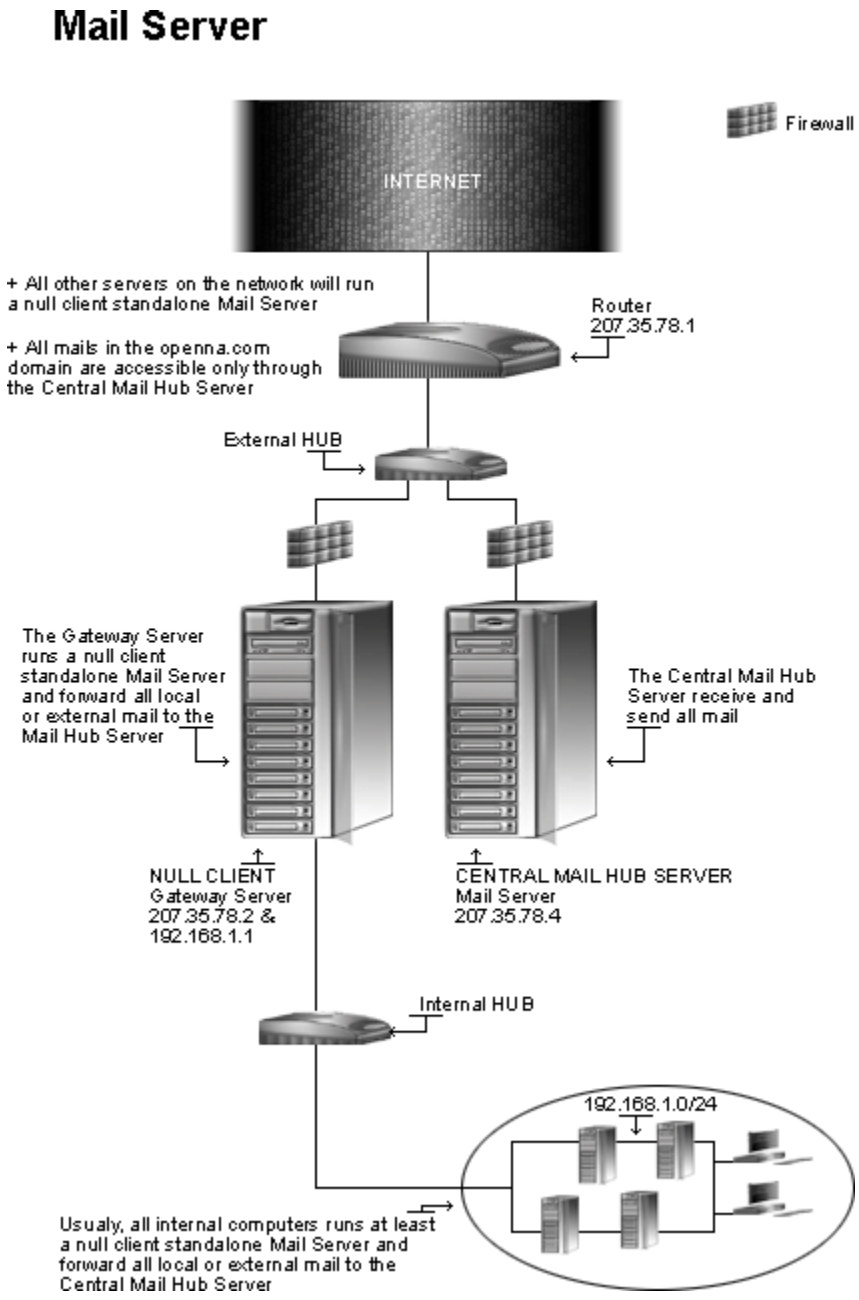
In our installation we'll provide you with two different configurations that you can set up for Exim; One for a Central Mail Hub Server, and another for a Null Client Mail Server.

The Central Mail Hub Server configuration will be used for your server where the assigned task is to send, receive and relay all mail for all local, client and server mail machines you may have on your network. A Null Client Mail Server refers to all other local server or client machines on your network that run Exim to send all mail to the Central Mail Hub for future delivery.

You can configure Exim so that it accepts only mail that is generated locally, thus insulating neighbor machines for easier security. This kind of client never receives mail directly via the Internet; instead, all mail from the Internet for those computers is kept on the Mail Hub server. It is a good idea to run at least one Central Mail Hub Server for all computers on your network; this architecture will limit the management tasks on the server and client machines, and will greatly improve the security of your site.

If you decide to install and use Exim as your Central Mail Hub Server, it will be important to refer to the parts that talk about **I**nternet **M**essage **A**ccess **P**rotocol in this book. Recall that Exim is just a program to send and receive mail and cannot be used to read mail. Therefore, in a Central Mail Hub environment, you need to have a program which allows users to connect to the Exim Mail Hub to get and read their mail, this is where a program like UW IMAP, Tpop3d, or Qpopper, also know as a **I**nternet **M**essage **A**ccess **P**rotocol (IMAP) or **P**ost **O**ffice **P**rotocol (POP) is required and must be installed if you run Exim as your Mail Hub Server and ONLY in this case.

If you run Exim as a Null Client Mail Server, then you don't need to install an Internet Message Access Protocol like UW IMAP, Tpop3d, or Qpopper. If you decide to skip this chapter about Exim because you'd prefer to install Qmail as your MTA, then you don't need to install UW IMAP, Tpop3d, or Qpopper even if you configure Qmail as a Mail Hub Server since Qmail already come with its own fast, small and secure POP program know as qmail-popd3.



This is a graphical representation of the `Mail` Server configuration we use in this book. We try to show you different settings (Central Mail Hub Server, and Null Client Mail Server) on different servers. Lots of possibilities exist, and depend on your needs and network architecture.

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest Exim version number is 4.05

The procedures given in this chapter are likely to work on all Linux platforms, but we have only tested it on OpenNA Linux and Red Hat Linux.

Packages

The following is based on information listed by Exim as of 2002/06/24. Please check http://www.exim.org/ regularly for the latest status. We chose to install from source because it provides the facility to fine tune the installation.

Source code is available from:
Exim Homepage: http://www.exim.org/
You must be sure to download: exim-4.05.tar.gz

Prerequisites

Exim requires that the listed software below be already installed on your system to be able to compile successfully. If this is not the case, you must install it from your Linux CD-ROM or source archive files. Please make sure you have this program installed on your machine before you proceed with this chapter.

- ✓ OpenSSL is required to run Exim with SSL support on your system.

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all the files installed onto the system if you want to update the package in the future. To solve this problem, it's a good idea to make a list of files on the system before you install Exim, and then one afterwards, and then compare them using the **diff** utility to find out what files were placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep root]# find /* > Exim1
  ```
- And the following one after you install the software:
  ```
  [root@deep root]# find /* > Exim2
  ```
- Then use the following command to get a list of what changed:
  ```
  [root@deep root]# diff Exim1 Exim2 > Exim-Installed
  ```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. In our example above, we use the /root directory of the system to store all the generated file lists.

Compiling - Optimizing & Installing Exim

Below are the steps that you must make to configure, compile and optimize the Exim software before installing it on your system. First off, we install the program as user 'root' so as to avoid any authorization problems.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- This can be done with the following commands:

```
[root@deep /]# cp exim-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf exim-version.tar.gz
```

Step 2
Exim needs a UID and GID to properly run on the system but this UID/GID cannot run as super-user root; for this reason we must create a special user with no shell privileges on the system for running Exim daemon.

- To create this special Exim user on OpenNA Linux, use the following command:

```
[root@deep tmp]# groupadd -g 12 mail > /dev/null 2>&1 || :
[root@deep tmp]# useradd -c "Mail Server" -d /var/spool/mqueue -g 12 -s
/bin/false -u 8 mail > /dev/null 2>&1 || :
```

- To create this special Exim user on Red Hat Linux, use the following command:

```
[root@deep tmp]# groupadd -g 12 mail > /dev/null 2>&1 || :
[root@deep tmp]# useradd -u 8 -g 12 -s /bin/false -M -r -d
/var/spool/mqueue mail > /dev/null 2>&1 || :
```

The above command will create a null account, with no password, no valid shell, no files owned-nothing but a UID and a GID for the program. Remember that Exim daemon does not need to have a shell account on the server.

WARNING: On much Linux system like Red Hat Linux, the UID and GID "mail" already exist, therefore check inside your /etc/passwd and /etc/group files before creating the above user and group on your server. On OpenNA Linux, you have to create it.

Step 3
Now, edit the **shells** file (vi /etc/shells) and add a non-existent shell name "/bin/false", which is the one we used in the useradd command above.

```
[root@deep tmp]# vi /etc/shells
/bin/bash2
/bin/bash
/bin/sh
/bin/false ← This is our added no-existent shell
```

Step 4
After that, move into the newly created Exim directory and perform the following steps before compiling and optimizing it. The modifications and configurations we bring to the Exim source files below are necessary to relocate some of the default files and programs, make the Exim software server run faster, as well as to be compatible with our Linux operating system.

- To move into the newly created Exim directory, use the following command:
 [root@deep tmp]# **cd exim-4.05/**

Step 5
Exim use a different procedure to install in the system, instead of using the default GNU autoconf build like many open source program use, it go with a file called EDITME which allow it to compile an appropriate Makefile for your specific system. Therefore, we have to copy the file called src/EDITME in a new file called Local/Makefile and move into this directory to configure the program for our server.

- This can be done with the following commands.
 [root@deep exim-4.05]# **cp src/EDITME Local/Makefile**

Step 6
Once the file called EDITME has been copied into the Local directory and renamed "Makefile", we can edit it to configure the software for our own needs and our operating system.

The Local/Makefile is the main build-time configuration file for Exim. It is in this file that we define all settings that we need to have with Exim. Below we show you all default settings that you should change to make Exim work on your Linux system.

- Edit the **Makefile** file (vi Local/Makefile), and change all of the following lines.

 BIN_DIRECTORY=/usr/exim/bin

 To read:

 BIN_DIRECTORY=/usr/sbin

 Here we define where we want the exim binary to be installed on our system.

 CONFIGURE_FILE=/usr/exim/configure

 To read:

 CONFIGURE_FILE=/etc/mail/exim.conf

 Here we define where Exim's run time configuration file (exim.conf) is to be found and which name it should have. The name of the file is compiled into the binary for security reasons.

 EXIM_USER=

 To Read:

 EXIM_USER=8

Here we define the UID under which we want Exim to run on the system. The UID "8" correspond to the username "mail" we have previously created.

```
# EXIM_GROUP=
```

To Read:

```
EXIM_GROUP=12
```

Here we define the GID under which we want Exim to run on the system. The GID "12" correspond to the group name "mail" we have created previously.

```
SPOOL_DIRECTORY=/var/spool/exim
```

To Read:

```
SPOOL_DIRECTORY=/var/spool/mqueue
```

Here we define the directory where all the data for messages in transit are kept.

```
# SUPPORT_MAILDIR=yes
```

To Read:

```
SUPPORT_MAILDIR=yes
```

Here we uncomment the SUPPORT_MAILDIR option to enable support for Maildir format with Exim.

```
EXIM_MONITOR=eximon.bin
```

To Read:

```
# EXIM_MONITOR=eximon.bin
```

Here we comment out the EXIM_MONITOR option to disable support for Exim Monitor. Exim Monitor requires an X11 display to work and X11 is not installed on our secure server, therefore we can safety disable this option.

```
# AUTH_CRAM_MD5=yes
# AUTH_PLAINTEXT=yes
```

To Read:

```
AUTH_CRAM_MD5=yes
AUTH_PLAINTEXT=yes
```

Here we uncomment the AUTH_CRAM_MD5 and AUTH_PLAINTEXT options to enable support for SMTP authentication of both protocols. These authenticators are included into the binary for security reasons.

```
# SUPPORT_TLS=yes
# TLS_LIBS=-lssl -lcrypto
```

To Read:

```
SUPPORT_TLS=yes
TLS_LIBS=-lssl -lcrypto
```

Here we uncomment the SUPPORT_TLS and TLS_LIBS options to enable support for SMTP with SSL encryption support. If you don't want to provide SSL support with Exim, you can keep the above lines commented out. You can enable SSL support with Exim even if you don't use it and Exim will still work.

```
# LOG_FILE_PATH=syslog
```

To Read:

```
LOG_FILE_PATH=syslog
```

Here we uncomment the LOG_FILE_PATH option to enable support for syslog with Exim and inform Exim to logs all information into the syslog facility of our system.

```
ZCAT_COMMAND=/usr/bin/zcat
```

To Read:

```
ZCAT_COMMAND=/usr/bin/gunzip
```

Here we define the location of the command we want to use to allow Exim to uncompress files on our system when required.

```
# EXIM_PERL=perl.o
```

To Read:

```
EXIM_PERL=perl.o
```

Here we uncomment the EXIM_PERL option to enable support for Perl with Exim. We need Perl to be able to use Anti-Virus and Anti-Spam features with Exim.

```
# CHOWN_COMMAND=/usr/bin/chown
# CHGRP_COMMAND=/usr/bin/chgrp
```

To Read:

```
CHOWN_COMMAND=/bin/chown
CHGRP_COMMAND=/bin/chgrp
```

Here we uncomment the CHOWN_COMMAND and CHGRP_COMMAND options to define the location of these commands on our system.

```
# SUPPORT_MOVE_FROZEN_MESSAGES=yes
```

To Read:

```
SUPPORT_MOVE_FROZEN_MESSAGES=yes
```

Here we uncomment the `SUPPORT_MOVE_FROZEN_MESSAGES` option to inform `Exim` to automatically move frozen messages out of the main spool directory when required.

Step 7
Next, we have to edit the `Makefile-Linux` file and define optimization `FLAGS` specific to our `CPU` architecture that we want to use to compile `Exim` on our system.

- Edit the **`Makefile-Linux`** file (`vi OS/Makefile-Linux`), and change the line.

```
CFLAGS=-O
```

To read:

```
CFLAGS=-O2 -march=i686 -funroll-loops
```

Step 8
Now, we must make a list of all files on the system before installing the software, and one afterwards, then compare them using the **`diff`** utility to find out what files are placed where and finally we install the `Exim` software.

```
[root@deep exim-4.05]# make
[root@deep exim-4.05]# cd
[root@deep root]# find /* > Exim1
[root@deep root]# cd /var/tmp/exim-4.05/
[root@deep exim-4.05]# make install
[root@deep exim-4.05]# ln -fs /usr/sbin/exim-4.05-1 /usr/lib/sendmail
[root@deep exim-4.05]# ln -fs /usr/sbin/exim-4.05-1 /usr/sbin/sendmail
[root@deep exim-4.05]# ln -fs /usr/sbin/exim-4.05-1 /usr/bin/mailq
[root@deep exim-4.05]# ln -fs /usr/sbin/exim-4.05-1 /usr/bin/runq
[root@deep exim-4.05]# mv /etc/aliases /etc/mail/
[root@deep exim-4.05]# strip /usr/sbin/exim-4.04-1
[root@deep exim-4.05]# chown 0.mail /var/spool/mail/
[root@deep exim-4.05]# chmod 1777 /var/spool/mail/
[root@deep exim-4.05]# cd
[root@deep root]# find /* > Exim2
[root@deep root]# diff Exim1 Exim2 > Exim-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

Step 9
Once the configuration, optimization, compilation, and installation of the Exim software have been accomplished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete Exim and its related source directory, use the following commands:
```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf exim-version/
[root@deep tmp]# rm -f exim-version.tar.gz
```

The rm command as used above will remove all the source files we have used to compile and install Exim. It will also remove the Exim compressed archive from the /var/tmp directory.

Configuring Exim

After Exim has been built and installed successfully in your system, your next step is to configure and customize its configuration files to fit your needs.

- ✓ /etc/mail/exim.conf: (The Exim Configuration File)
- ✓ /etc/mail/localdomains: (The Exim Local Configuration File)
- ✓ /etc/mail/relaydomains: (The Exim Relay Configuration File)
- ✓ /etc/mail/aliases: (The Exim aliases File)
- ✓ /etc/mail/access: (The Exim Access Configuration File)
- ✓ /etc/mail/system-filter: (The Exim System Filter File)
- ✓ /etc/sysconfig/exim: (The Exim System Configuration File)
- ✓ /etc/init.d/exim: (The Exim Initialization File)

/etc/mail/exim.conf: The Exim Configuration File

The /etc/mail/exim.conf file is the main configuration file for Exim. It is in this configuration file that Exim gets not just all of its information, but features to enable, disable, the name and location of different files to use, the domain or server for which it is responsible, and so forth.

The exim.conf file has several customizable options and declarations available depending on the type of Exim service that you want to offer. Here are the most important parameters to configure for maximum security; a complete listing and/or special requirements are available on the Exim web site. We must configure the most important ones to suit our requirements.

There are myriad of declarations, options and parameter available with Exim. These may or may not be required depending of the type of Exim server that you want for your network. In the configuration below, we cover most of the important parameters for a secure SMTP server that can be easily adjusted to fit a more complex network environment if required.

The goal for a secure configuration file is to limit the complexity of the file to avoid errors induced by poor design and implementation.

A typical exim.conf file beginning with:

1) A "**Main Configuration**" part common for the entire configuration file;

2) An "**ACL Configuration**" part for controlling incoming SMTP mail;

3) A "**Routers Configuration**" part which process addresses and determine how the message is to be delivered;

4) A "**`Transports Configuration`**" part which defines mechanisms for copying messages to destinations;

5) A "**`Retry Configuration`**" rules, for use when a message cannot be immediately delivered;

6) A "**`Rewrite Configuration`**" rules, for use when a message arrives and when new addresses are generated during delivery;

7) An "**`Authenticator Configuration`**" part used by the `SMTP AUTH` command for relaying feature.

- Edit the **`exim.conf`** file (`vi /etc/mail/exim.conf`). Below is what we recommend you set:

```
######################################################################
#                    MAIN CONFIGURATION SETTINGS                     #
######################################################################

primary_hostname = smtp.domain.com

acl_smtp_rcpt = check_recipient
acl_smtp_data = check_message

domainlist local_domains = @ : lsearch;/etc/mail/localdomains
hostlist relay_hosts = lsearch;/etc/mail/relaydomains
hostlist auth_relay_hosts = *

log_selector =  \
        +all_parents \
        +received_sender \
        +received_recipients \
        +smtp_confirmation \
        +smtp_syntax_error

allow_domain_literals = false
never_users = root:daemon:bin:sync:named
host_lookup = *
trusted_users = mail
gecos_pattern = ^([^,:]*)
gecos_name = $1
freeze_tell = postmaster
auto_thaw = 1h
ignore_bounce_errors_after = 30m
timeout_frozen_after = 7d

received_header_text = "Received: \
        ${if def:sender_rcvhost {from ${sender_rcvhost}\n\t}\
        {${if def:sender_ident {from ${sender_ident} }}\
        ${if def:sender_helo_name {(helo=${sender_helo_name})\n\t}}}}\
        by ${primary_hostname} \
        ${if def:received_protocol {with ${received_protocol}}} \
        (Exim ${version_number} #${compile_number} (OpenNA Linux))\n\t\
        id ${message_id}\
        ${if def:received_for {\n\tfor <$received_for>}}"

system_filter = /etc/mail/system-filter
message_body_visible = 5000
message_size_limit = 10M
smtp_accept_max = 2048
smtp_connect_backlog = 256
```

```
queue_only
split_spool_directory
queue_run_max = 1
remote_max_parallel = 1
rfc1413_hosts = *
rfc1413_query_timeout = 0s

smtp_banner = "Welcome on our mail server!\n\
        This system does not accept Unsolicited \
        Commercial Email\nand will blacklist \
        offenders via our spam processor.\nHave a \
        nice day!\n\n${primary_hostname} ESMTP Exim \
        ${version_number} ${tod_full}"

#############################################################################
#                       ACL CONFIGURATION                                   #
#         Specifies access control lists for incoming SMTP mail             #
#############################################################################

begin acl

check_recipient:
  accept  hosts = :

  deny    local_parts   = ^.*[@%!/|]

  deny    senders       = *@dbm;/etc/mail/access.db : \
                          dbm;/etc/mail/access.db

  require verify        = sender

  deny    message       = unrouteable address
          hosts         = !127.0.0.1/8:0.0.0.0/0
         !verify        = recipient

  accept  domains       = +local_domains
          endpass
          message       = unknown user
          verify        = recipient

  accept  hosts         = +relay_hosts

  accept  hosts         = +auth_relay_hosts
          endpass
          message       = authentication required
          authenticated = *

  deny    message       = relay not permitted


check_message:
  accept

#############################################################################
#                      ROUTERS CONFIGURATION                                #
#               Specifies how addresses are handled                         #
#############################################################################
#     THE ORDER IN WHICH THE ROUTERS ARE DEFINED IS IMPORTANT!               #
# An address is passed to each router in turn until it is accepted.         #
#############################################################################

begin routers
```

```
dnslookup:
  driver = dnslookup
  domains = ! +local_domains
  transport = remote_smtp
  ignore_target_hosts = 0.0.0.0 : 127.0.0.0/8
  no_more

system_aliases:
  driver = redirect
  allow_fail
  allow_defer
  data = ${lookup{$local_part}lsearch{/etc/mail/aliases}}
  user = mail
  file_transport = address_file
  pipe_transport = address_pipe

userforward:
  driver = redirect
  check_local_user
  file = $home/.forward
  no_verify
  no_expn
  check_ancestor
  allow_filter
  modemask = 002
  file_transport = address_file
  pipe_transport = address_pipe
  reply_transport = address_reply

localuser:
  driver = accept
  check_local_user
  transport = local_delivery

######################################################################
#                     TRANSPORTS CONFIGURATION                       #
######################################################################
#                       ORDER DOES NOT MATTER                        #
#     Only one appropriate transport is called for each delivery.    #
######################################################################

begin transports

remote_smtp:
  driver = smtp

local_delivery:
  driver = appendfile
  file = /var/mail/$local_part
  delivery_date_add
  envelope_to_add
  return_path_add
  group = mail
  mode = 0600

address_pipe:
  driver = pipe
  return_output

address_file:
  driver = appendfile
  delivery_date_add
  envelope_to_add
```

```
  return_path_add

address_reply:
  driver = autoreply

#########################################################################
#                       RETRY CONFIGURATION                             #
#########################################################################

begin retry

# Domain               Error       Retries
# ------               -----       -------

*                      *           F,2h,15m; G,16h,1h,1.5; F,4d,6h

#########################################################################
#                       REWRITE CONFIGURATION                           #
#########################################################################

begin rewrite

#########################################################################
#                   AUTHENTICATION CONFIGURATION                        #
#########################################################################

begin authenticators
```

This tells the `exim.conf` file to set itself up for this particular configuration with:

Main Configuration

`primary_hostname = smtp.domain.com`
This configuration option is used to specify the fully qualified "official" name of your host on which the mail server is running. For the proper functionality of `Exim`, it is absolutely important to fill this option with the `FQDN` of your system. In our example, we use smtp.domain.com, don't forget to change it for the `FQDN` of your system or it will fail to work. This option is possibly the only one you really need to change to make the `Exim` configuration file to work on your server.

```
acl_smtp_rcpt = check_recipient
acl_smtp_data = check_message
```
The above configuration options are used to define the name of the `ACL` used later in this `Exim` configuration file to control incoming messages. You should not change this setting. There is six different types of `ACL` that we can use with `Exim` as follows.

- `acl_smtp_auth` specifies the `ACL` to run when `AUTH` is received.
- `acl_smtp_data` specifies the `ACL` to run after a message has been received.
- `acl_smtp_etrn` specifies the `ACL` to run when `ETRN` is received.
- `acl_smtp_expn` specifies the `ACL` to run when `EXPN` is received.
- `acl_smtp_rcpt` specifies the `ACL` to run when `RCPT` is received.
- `acl_smtp_vrfy` specifies the `ACL` to run when `VRFY` is received.

In general and in most cases, we only need to use the "`RCPT`" and "`DATA`" `ACL` with `Exim`. Other `ACL's` are for advanced features or configurations and are not required for proper operation of a mail server.

```
domainlist local_domains = @ : lsearch;/etc/mail/localdomains
hostlist relay_hosts = lsearch;/etc/mail/relaydomains
hostlist auth_relay_hosts = *
```

The above settings are used to create and define files that will be used to allow or deny relaying with the SMTP server. By default Exim does not allow any external hosts or domains to relay. The above options define the files on which we will list hosts or domains allowed to relay through our mail server. These settings will be referred to in more detail later in our Exim ACL configuration.

The first setting "domainlist local_domains" is used to specify your local domains on which Exim is running and generally, you don't need to change it. The "@" means the name of the local host (www.domain.com) and the "lsearch" redirect the parameter to an additional file to read for the information. If you don't want to do any local deliveries, you can remove the "@" from the setting above. This should be required only when you want to configure Exim as a Null Client Mail Server.

The second setting "hostlist relay_hosts" is used to specify hosts that can use your Exim Mail Server as an outgoing relay to any other host on the Internet. Such a setting commonly refers to a complete local network as well as the localhost. Don't be confused here; you DON'T need to list any systems that interact with your Exim mail server but just those servers on your network that need to send mails externally (the Internet).

Again, we use the "lsearch" macro in this parameter line to indicate to Exim to get its information through the file called "relaydomains" located under the /etc/mail directory. This allows us to list all hosts in this file instead of using our exim.conf file for this purpose.

A perfect good example is as follows:
Exim is installed as a Central Mail Hub Server on smtp.domain.com (your mail server machine).
Exim is installed as a Null Client Mail Server on www.domain.com (your web server machine).
Your web server from time to time needs to send mail externally (the Internet) because you use some kind of mail form or electronic commerce.

Therefore you will add www.domain.com into your "relaydomains" file to allow it to use Exim on the Central Mail Hub Server for relaying to the external network (the Internet).

A bad example is as follows:
Exim is installed as a Central Mail Hub Server on smtp.domain.com (your mail server machine).
Exim is installed as a Null Client Mail Server on ns1.domain.com (your dns1 server machine).
Your DNS1 server from time to time needs to send locally generated mail to the Central Mail Hub Server for delivery but NOT to the external (the Internet).

Therefore you DON'T need to add ns1.domain.com into your "relaydomains" file to allow it to use Exim on the Central Mail Hub Server because it doesn't need to relay externally but just delivers its message to the Mail Hub Server.

```
log_selector =  \
        +all_parents \
        +received_sender \
        +received_recipients \
        +smtp_confirmation \
        +smtp_syntax_error
```

The parameters above are used to define logging options that we want to use with Exim. In our configuration we log everything related to the mail server. This means that if you send, receive, forward, etc mails, then all actions will be logged to the /var/log/maillog file for verification. This is a security feature.

```
allow_domain_literals = false
```
This configuration option is used to prevents Exim from recognizing addresses of the form "user@[111.111.111.111]" that is, with a "domain literal" (an IP address) instead of a named domain that crackers could use to relay mails with your server. This is a security feature to protect your mail server for unwanted relaying.

```
never_users = root:daemon:bin:sync:named
```
This configuration option is used to list all local users from which no deliveries will ever be run. This means that all users listed in this setting will not be allowed to receive mail. In our example, you cannot deliver mail addressed to root, daemon, bin, sysnc, or named. This is not a problem since most sites have an alias for those users that redirects such mail to a human administrator. You can add to the above list any system user's accounts that you want with a colon-separated list. This is a security feature.

```
host_lookup = *
```
This configuration option is used to inform Exim to do a reverse DNS lookup on all incoming mails, in order to get the true host name. This penalizes SMTP performance and on highly loaded mail servers, I recommend you to disable this setting by removing the line. This is a performance feature.

```
trusted_users = mail
```
When this configuration option is used, then any process that is running as one of the listed users may pass a message to Exim and specify the sender's address using the "-f" command line option, without Exim's adding a "Sender" header. In general, we need this option for virus scanners or spam software to run with Exim. You can add to the above list any users that you want with a colon-separated list. This is a security feature.

```
gecos_pattern = ^([^,:]*)
gecos_name = $1
```
Some operating systems use the "gecos" field of mailer software in the system password file to hold other information in addition to users' real names. Exim looks up this field when it is creating "Sender" and "From" headers. If these options are set, Exim uses "gecos_pattern" to parse the gecos field, and then expands "gecos_name" as the user's name. This is a security feature to limit spam again.

```
freeze_tell = postmaster
```
This configuration option is used to send a mail to the specified system user account "postmaster" when a message is frozen. There are many reasons for messages to be frozen; one is if Exim cannot deliver a mail with no return address (normally a bounce), another that may be common on dialup system, is if a DNS lookup of a smarthost fails. In any case it is good to define and use this option to be informed when frozen messages are on the queue since this happen often with all the garbage spammers send to the Internet.

```
auto_thaw = 1h
```
This configuration option is used to inform Exim to try a new delivery attempt on any frozen messages if this much time has passed since it was frozen. In our configuration, we set the retry time to one hour. A good strategy is to use a lower time like we do (1h) to avoid spammers' messages to stay for a long period of time in the queue.

```
ignore_bounce_errors_after = 30m
```
This configuration option is used to unfreeze bounce messages after the specified period of time (30m), tries once more to deliver them, and ignores any delivery failures. This is one of the Exim features that you will see often on mail server when messages cannot be delivered. It is a good idea to change the default setting of "2d" for "30m".

```
timeout_frozen_after = 7d
```
Exim uses this configuration option to cancel (remove) frozen messages that are older than a week (`7d`).

```
received_header_text = "Received: \
        ${if def:sender_rcvhost {from ${sender_rcvhost}\n\t}\
        {${if def:sender_ident {from ${sender_ident} }}\
        ${if def:sender_helo_name {(helo=${sender_helo_name})\n\t}}}}\
        by ${primary_hostname} \
        ${if def:received_protocol {with ${received_protocol}}} \
        (Exim ${version_number} #${compile_number} (OpenNA Linux))\n\t\
        id ${message_id}\
        ${if def:received_for {\n\tfor <$received_for>}}"
```
This string defines the contents of the "`Received`" message header that is added to each message, except for the timestamp, which is automatically added on at the end, preceded by a semicolon. The string is expanded each time it is used.

```
system_filter = /etc/mail/system-filter
```
This configuration option is used to specify a filter file, which is applied to all messages before any routing or directing is done. This is called the "`system message filter`" and we use it to better control the security and filtering features for our mail server. In our configuration, we redirect the option to a file called "`system-filter`" located under the `/etc/mail` directory which handles all of our filtering parameters.

```
message_body_visible = 5000
```
This configuration option is used to specify how much of a message's body is to be included in the `message_body` expansion variable. Default value is `500`, but we need to increase it if we use the "`message_filter`" option above.

```
message_size_limit = 10M
```
This configuration option is used to limit the maximum size of message that `Exim` will be allowed to process. Incoming `SMTP` messages are failed with a `552` error if the limit is exceeded. In our configuration, we limit the size of messages that could be sending or received by `Exim` to 10 MB.

```
smtp_accept_max = 2048
```
This configuration option is used to specify the maximum number of simultaneous incoming `SMTP` calls that `Exim` will accept. On busy mail server, the above value if fine but on small mail server, you should lower the values to something like "`512`" to avoid possible `DoS` attacks. This is both a performance and security feature.

```
smtp_connect_backlog = 256
```
This configuration option is used to specify the maximum number of waiting `SMTP` connections. `Exim` passes this value to the `TCP/IP` system when it sets up its listener. Once these numbers of connections are waiting for the daemon's attention, subsequent connection attempts are refused at the `TCP/IP` level. This is a performance feature.

```
queue_only
```
When the above configuration option is set, a delivery process is not automatically started whenever a message is received. Instead, the message waits on the queue for the next queue run. This is a performance feature when mixed with the options below.

split_spool_directory
This configuration option is used to cause Exim to split its input directory into 62 subdirectories, each with a single alphanumeric character as its name. The sixth character of the message id is used to allocate messages to subdirectories; this is the least significant base-62 digit of the time of arrival of the message. Splitting up the spool in this way may provide better performance on systems where there are long mail queues, by reducing the number of files in any one directory. This is a performance feature.

queue_run_max = 1
This configuration option is used to control the maximum number of queue-runner processes that an Exim daemon can run simultaneously. In our configuration, we set it to "1". This is a performance feature when mixed with the options below.

remote_max_parallel = 1
This configuration option is used to control parallel delivery to remote sites. If the value is less than 2, parallel delivery is disabled, and Exim does all the remote deliveries for a message one by one, from a single delivery process as other MTA's do. Otherwise, if a message has to be delivered to more than one remote host, or if several copies have to be sent to the same remote host, then up to remote_max_parallel deliveries are done simultaneously, each in a separate process.

If more than remote_max_parallel deliveries are required, the maximum numbers of processes are started, and as each one finishes, another is begun. Because each queue runner delivers only one message at a time, the maximum number of deliveries that can then take place at once is queue_run_max multiplied by remote_max_parallel. This option mixed with the above options (queue_only, split_spool_directory, and queue_run_max) will greatly improve message delivery performance if the Exim queue is made to run at each minute (-q1m) as we do in our setup. This is a performance feature.

rfc1413_hosts = *
rfc1413_query_timeout = 0s
The above configuration options cause Exim to make RFC 1413 (ident) callbacks for all incoming SMTP connections.

The first setting "rfc1413_hosts" is used to list the hosts to which these calls are made. The "*" option means make RFC 1413 (ident) callbacks for all incoming SMTP connections.

The second setting "rfc1413_query_timeout" define the timeout to use. If you set the timeout to zero (as we do), then all RFC 1413 calls are disable. It is highly recommended to avoid delays on starting up an SMTP session. This is a performance feature.

```
smtp_banner = "Welcome on our mail server!\n\
        This system does not accept Unsolicited \
        Commercial Email\nand will blacklist \
        offenders via our spam processor.\nHave a \
        nice day!\n\n${primary_hostname} ESMTP Exim \
        ${version_number} ${tod_full}"
```
This configuration option is used to simply implement customized SMTP welcome banner.

ACL Configuration

```
begin acl
```
Remember that each new Exim configuration part other than the first (Main Configuration) is introduced by the word "begin" followed by the name of the part, which is in our case "acl" to indicate the beginning of the **A**ccess **C**ontrol **L**ists part of the configuration.

The "ACL Configuration" part of Exim is used to define access control lists to use for all incoming SMTP mail on the server.

For more information about "ACL Configuration" with Exim, please visit:
http://www.exim.org/exim-html-4.00/doc/html/spec_37.html#CHAP37

```
check_recipient:
```
The above setting is the one we defined earlier during our Exim configuration. We use it here to inform the system that we want to start the ACL relating to every RCPT command in an incoming SMTP message. The tests are run in order until the address is either accepted or denied.

```
accept  hosts = :
```
This ACL allows Exim to accept mail only if the source is local SMTP (i.e. not over TCP/IP). We do this by testing for an empty sending host field.

```
deny    local_parts   = ^.*[@%!/|]
```
This ACL allows Exim to deny mail if the local part contains @ or % or / or | or !. These are rarely found in genuine local parts, but are often tried by people looking to circumvent relaying restrictions.

```
deny    senders       = *@dbm;/etc/mail/access.db : \
                        dbm;/etc/mail/access.db
```
This ACL allows Exim to deny any email addresses listed in the access file located under the /etc/mail directory. We can use the access file to list all email we want to block. The above parameter gets its information from the database format (.db) of the access file for better performance.

```
require verify        = sender
```
This ACL allows Exim to deny mail unless the sender address can be verified.

```
deny    message       = unrouteable address
        hosts         = !127.0.0.1/8:0.0.0.0/0
       !verify        = recipient
```
This ACL denies Exim accepting mail except if the address is our localhost. It also informs Exim to not verify the recipient of the localhost.

```
accept  domains       = +local_domains
        endpass
        message       = unknown user
        verify        = recipient
```
This ACL allows Exim to accept mail if the address is in a local domain, but only if the recipient can be verified. Otherwise deny. The "endpass" line is the border between passing on to the next ACL statement (if tests above it fail) or denying access (if tests below it fail).

```
accept  hosts          = +relay_hosts
```
This ACL allows Exim to accept mail if the message comes from one of the hosts for which we are an outgoing relay.

```
accept  hosts          = +auth_relay_hosts
        endpass
        message        = authentication required
        authenticated = *
```
This ACL allows Exim to accept mail if the message arrived over an authenticated connection, from any host.

```
deny    message        = relay not permitted
```
Reaching the end of the ACL causes a "deny", but we might as well give an explicit message. Here is what we do.

```
check_message:
```
The above setting is the second we have defined earlier during our Exim configuration. We use it here to inform the system that we want to start ACL related to every DATA command in an incoming SMTP message. The tests are run in order until the address is either accepted or denied.

```
  accept
```
This ACL allows Exim to accept mail once messages have been filtered, approved and received by the above "check_recipient" ACL.

Routers Configuration

```
begin routers
```
As we supposed to know now, each new Exim configuration section, other than the first, (Main Configuration) is introduced by the word "begin" followed by the name of the section, which is here "routers" to indicate the beginning of the "Routers Configuration" section of this file.

The "Router Configuration" section of Exim is used to specify how addresses are handled. This means that routers process addresses and determine how the message is to be delivered.

For more information about "Router Configuration" with Exim, please visit:
http://www.exim.org/exim-html-4.00/doc/html/spec_14.html#CHAP14

```
dnslookup:
  driver = dnslookup
  domains = ! +local_domains
  transport = remote_smtp
  ignore_target_hosts = 0.0.0.0 : 127.0.0.0/8
  no_more
```
This router routes addresses that are not in local domains by doing a DNS lookup on the domain name. Any domain that resolves to 0.0.0.0 or to a loopback interface address (127.0.0.0/8) is treated as if it had no DNS entry.

```
system_aliases:
  driver = redirect
  allow_fail
  allow_defer
  data = ${lookup{$local_part}lsearch{/etc/mail/aliases}}
  user = mail
  file_transport = address_file
  pipe_transport = address_pipe
```

This router handles aliasing using a traditional `/etc/mail/aliases` file by checking whether the local part is defined as an alias in the `/etc/mail/aliases` file, and if so, redirects it according to the data that it looks up from that file.

```
userforward:
  driver = redirect
  check_local_user
  file = $home/.forward
  no_verify
  no_expn
  check_ancestor
  allow_filter
  modemask = 002
  file_transport = address_file
  pipe_transport = address_pipe
  reply_transport = address_reply
```

This router handles forwarding using traditional `.forward` files in users' home directories by checking for possible forwarding data set up by individual users. The file called `.forward` in the user's home directory is consulted. If it does not exist, or is empty, the router declines. Otherwise, the contents of `.forward` are interpreted as redirection data.

```
localuser:
  driver = accept
  check_local_user
  transport = local_delivery
```

This router matches local user mailboxes by delivering to local mailboxes, provided that the local part is the name of a local login, by accepting the address and queuing it for the `local_delivery` transport. Otherwise, we have reached the end of the routers, so the address is bounced.

Transports Configuration

```
begin transports
```

Each new `Exim` configuration section other than the first (`Main Configuration`) is introduced by the word "`begin`" followed by the name of the section, which is here "`transports`" to indicate the beginning of the "`Transports Configuration`" section of this file.

The "`Transports Configuration`" section of `Exim` is used to define mechanisms for copying messages to destinations. It is important to note that only one appropriate transport is called for each delivery.

For more information about "`Transports Configuration`" with `Exim`, please visit:
http://www.exim.org/exim-html-4.00/doc/html/spec_23.html#CHAP23

```
remote_smtp:
  driver = smtp
```
This transport is used for delivering messages over SMTP connections. All its options are defaulted. The list of remote hosts comes from the router.

```
local_delivery:
  driver = appendfile
  file = /var/mail/$local_part
  delivery_date_add
  envelope_to_add
  return_path_add
  group = mail
  mode = 0600
```
This transport is used for local delivery to user mailboxes in traditional BSD mailbox format.

```
address_pipe:
  driver = pipe
  return_output
```
This transport is used for handling pipe deliveries generated by alias or .forward files.

```
address_file:
  driver = appendfile
  delivery_date_add
  envelope_to_add
  return_path_add
```
This transport is used for handling deliveries directly to files that are generated by aliasing or forwarding.

```
address_reply:
  driver = autoreply
```
This transport is used for handling autoreplies generated by the filtering option of the userforward router.

Retry Configuration

```
begin retry
```
Each new Exim configuration section other than the first (Main Configuration) is introduced by the word "begin" followed by the name of the section, which is here "retry" to indicate the beginning of the “Retry Configuration” section of this file.

The “Retry Configuration” part of Exim is used when a message cannot be immediately delivered.

For more information about “Retry Configuration” with Exim, please visit:
http://www.exim.org/exim-html-4.00/doc/html/spec_31.html#CHAP31

```
*                       *           F,2h,15m; G,16h,1h,1.5; F,4d,6h
```
This single retry rule applies to all domains and all errors. It specifies retries every 15 minutes for 2 hours, then increasing retry intervals, starting at 1 hour and increasing each time by a factor of 1.5, up to 16 hours, then retries every 6 hours until 4 days have passed since the first failed delivery.

Rewrite Configuration

`begin rewrite`
Each new `Exim` configuration part other than the first (`Main Configuration`) is introduced by the word "`begin`" followed by the name of the part, which is here "`rewrite`" to indicate the beginning of the "`Rewrite Configuration`" part of this file.

The "`Rewrite Configuration`" part of `Exim` is used when a message arrives and when new addresses are generated during deliveries. In our configuration, we don't need to use it.

Authenticator Configuration

`begin authenticators`
Each new `Exim` configuration section other than the first (`Main Configuration`) is introduced by the word "`begin`" followed by the name of the section, which is here "`authenticators`" to indicate the beginning of the "`Authenticators Configuration`" section of this file.

The "`Authenticators Configuration`" section of `Exim` is used for `SMTP` authentication for relaying feature.

For more information about "`Authenticators Configuration`" with `Exim`, please visit:
http://www.exim.org/exim-html-4.00/doc/html/spec_32.html#CHAP32

Step2
Now, set the permission mode of the `exim.conf` file to be (`0640/-rw-r-----`) and owned by the super-user '`root`' with group permission set to "`mail`" user for security reasons.

- To change the permission mode and ownership of `exim.conf` file, use:
```
[root@deep /]# chmod 640 /etc/mail/exim.conf
[root@deep /]# chown 0.12 /etc/mail/exim.conf
```

`/etc/mail/localdomains`: The `Exim` Local Configuration File

The `/etc/mail/localdomains` file is read by `Exim` so it knows about all the domain names that are local to your network. If your domain name is "`domain.com`", you have to add it into this file for `Exim` to work.

You don't need to list all servers on your network into this file, just your domain name. Again, I repeat, there is no need to list "`www.domain.com`", or "`ftp.domain.com`", or "`something.domain.com`", etc into this file but ONLY your domain name "`domain.com`".

For virtual hosting, we will also use this file to list all virtual domains hosted on our mail server. See later in this chapter for more information about virtual domain hosting with `Exim`.

Step 1
By default, the localdomains file does not exist after the installation, we have to create it.

- Create the **localdomains** file (touch /etc/mail/localdomains) and add:

```
# localdomains - include all of your local domains name here.
# Virtual domains must be listed here to be recognized as local.
# N.B.: Exim must be restarted after this file is modified.
#
domain.com
```

Step2
Now, set the permission mode of the localdomains file to be (0640/-rw-r-----) and owned by the super-user 'root' with group permission set to "mail" user for security reasons.

- To change the permission mode and ownership of localdomains file, use:

```
[root@deep /]# chmod 640 /etc/mail/localdomains
[root@deep /]# chown 0.12 /etc/mail/localdomains
```

/etc/mail/relaydomains: The Exim Relay Configuration File

With Exim, relaying is denied by default (this is an Anti-Spam feature) and if you want to allow some domains in your network to relay through your mail server, you must create and use the "relaydomains" file to list each domain name allowed to relay through your Mail Server.

Step 1
By default, the relaydomains file does not exist after the installation, we have to create it.

- Create the **relaydomains** file (touch /etc/mail/relaydomains) and add:

```
# This file handle all domains from which relaying is allowed.
# By default we include the locahost of the server or nothing will work.
# Virtual Domains must be added to this list or relaying will be denied.
# N.B.: Exim must be restarted after this file is modified.
#
localhost
```

Step2
Now, set the permission mode of the relaydomains file to be (0640/-rw-r-----) and owned by the super-user 'root' with group permission set to "mail" user for security reasons.

- To change the permission mode and ownership of relaydomains file, use:

```
[root@deep /]# chmod 640 /etc/mail/relaydomains
[root@deep /]# chown 0.12 /etc/mail/relaydomains
```

/etc/mail/aliases: The Exim Aliases File

Aliasing in the mail server world is the process of converting one local recipients name on the system into another. Example uses are to convert a generic name (such as root) into a real username on the system, or to convert one name into a list of many names (for mailing lists).

Step 1
For security reasons, Exim never delivers mail to the super-user "root" and some minimal aliases are required by the mail RFCs 2821 and 2822 for mail servers to work. Therefore, we have to edit our default aliases file to configure some system user accounts to the address of a HUMAN who deals with system's mail problems.

- Edit the **aliases** file (vi /etc/mail/aliases) and add/change the following lines. Below is what we recommend you set.

```
# The following aliases are required by the mail RFCs 2821 and 2822.
# At least, you should set "postmaster" to the address of a HUMAN
# who deals with this system's mail problems.
#
postmaster:     sysadmin@domain.com
mailer-daemon:  postmaster
root:           postmaster

# It is a good idea to redirect any messages sent to system accounts
# so that they don't just get ignored.
#
bin:            root
daemon:         root
sync:           root
mail:           root
pop:            root
uucp:           root
ftp:            root
nobody:         root
www:            root
named:          root
postgres:       root
mysql:          root
squid:          root
amavis:         root
operator:       root
abuse:          root
hostmaster:     root
webmaster:      root
```

NOTE: Please, don't forget to change "postmaster" to the email address of your real system administrator on your network. The above "sysadmin@domain.com" is an example, therefore change it. Your aliases file will be probably far more complex, but even so, note how the example shows the minimum form of aliases.

Step2
Now, set the permission mode of the aliases file to be (0640/-rw-r-----) and owned by the super-user 'root' with group permission set to "mail" user for security reasons.

- To change the permission mode and ownership of "aliases" file, use the commands:

```
[root@deep /]# chmod 640 /etc/mail/aliases
[root@deep /]# chown 0.12 /etc/mail/aliases
```

Step 3
For every envelope that lists a local user as a recipient, Exim looks up that recipient's name in the "aliases" file. Because Exim may have to search through thousands of names in the "aliases" file, it's a good idea to create a copy of the file in a separate "db" database format file to significantly improve lookup speed.

A small program called "exim_dbmbuild" comes with Exim to achieve this. We can use it directly from the console each time we want to build/rebuild the "aliases" database or create a script file to automate the process. Below, we show you both methods.

- To manually build/rebuild the aliases database, use the following command:
```
[root@deep /]# cd /etc/mail/
[root@deep mail]# /usr/sbin/exim_dbmbuild aliases aliases.db
```

- To automate the building/rebuilding of the aliases database, create a script file called "newaliases" under the /usr/sbin directory.
```
[root@deep /]# cd /usr/sbin/
[root@deep sbin]# touch newaliases
[root@deep sbin]# chmod 510 newaliases
[root@deep sbin]# chown 0.0 newaliases
```

- Now edit the **newaliases** script (vi /usr/sbin/newaliases) and add the lines:
```
#!/bin/sh
/usr/sbin/exim_dbmbuild /etc/mail/aliases /etc/mail/aliases.db
/bin/chown root.mail /etc/mail/aliases
/bin/chmod 640 /etc/mail/aliases
/bin/chown root.mail /etc/mail/aliases.db
/bin/chmod 640 /etc/mail/aliases.db
```

NOTE: With the above "newaliases" script, you only need to run the script for the "aliases" database of Exim to be automatically rebuilt with the proper permissions and ownership.

/etc/mail/access: The Exim Access Configuration File

The /etc/mail/access file can be used to reject mail from selected email addresses. For example, you may choose to reject all mail originating from known spammers but it's probably better use SpamAssassin as described in this book for this purpose. Anyway, it is a good idea to have this file if we have some special names to put inside it.

In our configuration, we use this file to list all email addresses from which we don't want to accept mails. This is useful to block undesired mails coming in our mailbox.

Step 1
By default, the access file does not exist after the installation, we have to create it.

- Create the **access** file (touch /etc/mail/access) and add the following lines:

```
# The value part of the file must contain any email addresses from
# which you want to block access for sending mail to your server.
# N.B.: Exim must be restarted after this file is modified.
# Please list each email address one per line.
#
```

Step2
Now, set the permission mode of the access file to be (0640/-rw-r-----) and owned by the super-user 'root' with group permission set to "mail" user for security reasons.

- To change the permission mode and ownership of access file, use the commands:
```
[root@deep /]# chmod 640 /etc/mail/access
[root@deep /]# chown 0.12 /etc/mail/access
```

Step 3
For every incoming connection, Exim looks up the sender's email address in the "access" file. Because Exim may have to search through thousands of email in the "access" file, it's a good idea to create a copy of the file in a separate "db" database format file to significantly improve lookup speed.

A small program called "exim_dbmbuild" comes with Exim to archive this. We can use it directly from the console each time we want to build/rebuild the "access" database or create a script file to automate the process. Below, we show you both methods.

- To manually build/rebuild the access database, use the following command:
```
[root@deep /]# cd /etc/mail/
[root@deep mail]# /usr/sbin/exim_dbmbuild access access.db
```

- To automate the building/rebuilding of the access database, create a script file called "newaccess" under the /usr/sbin directory.
```
[root@deep /]# cd /usr/sbin/
[root@deep sbin]# touch newaccess
[root@deep sbin]# chmod 510 newaccess
[root@deep sbin]# chown 0.0 newaccess
```

- Now edit the **newaccess** script (vi /usr/sbin/newaccess) and add the lines:
```
#!/bin/sh
/usr/sbin/exim_dbmbuild /etc/mail/access /etc/mail/access.db
/bin/chown root.mail /etc/mail/access
/bin/chmod 640 /etc/mail/access
/bin/chown root.mail /etc/mail/access.db
/bin/chmod 640 /etc/mail/access.db
```

NOTE: With the above "newaccess" script, you only need to run the script for the "access" database of Exim to be automatically rebuilt with the proper permissions and ownership.

/etc/mail/system-filter: The Exim System Filter File

The /etc/mail/system-filter file is used by Exim to get information about how it should filter process or answer incoming or outgoing mails on the server. We use it to improve security of our mailer by defining rules that can detect buffer overruns, virus header, mime messages with suspicious name extension, VBS attachment and so on. In general it is a very good addition to our secure mail server.

Step 1
By default, the system-filter file does not exist after installation, we have to create it.

- Create the **system-filter** file (touch /etc/mail/system-filter) and add:

```
# Exim filter
#
## ------------------------------------------------------------------------
# Only run any of this stuff on the first pass through the filter - this
# is an optimization for messages that get queued and have several
# delivery attempts. We express this in reverse so we can just bail out
# on inappropriate messages.
#
if not first_delivery
then
  finish
endif

## ------------------------------------------------------------------------
# Check for MS buffer overruns as per BUGTRAQ.
# This could happen in error messages, hence its placing here...
# We substract the first n characters of the date header and test if its
# the same as the date header... which is a lousy way of checking if the
# date is longer than n chars long.
#
if ${length_80:$header_date:} is not $header_date:
then
  fail text "This message has been rejected because it has\n\
             an overlength date field which can be used\n\
             to subvert Microsoft mail programs\n\
             The following URL has further information\n\

http://www.securityfocus.com/frames/?content=/templates/article.html%3Fid
%3D61"
  seen finish
endif

## ------------------------------------------------------------------------
# These messages are now being sent with a <> envelope sender, but
# blocking all error messages that pattern match prevents bounces
# getting back.... so we fudge it somewhat and check for known
# header signatures. Other bounces are allowed through.
#
if $header_from: contains "@sexyfun.net"
then
  fail text "This message has been rejected since it has\n\
             the signature of a known virus in the header."
  seen finish
endif
if error_message and $header_from: contains "Mailer-Daemon@"
then
  # looks like a real error message - just ignore it
  finish
endif
```

```
## ----------------------------------------------------------------------
# Look for single part MIME messages with suspicious name extensions.
# Check Content-Type header using quoted filename
[content_type_quoted_fn_match]
#
if $header_content-type: matches
"(?:file)?name=(\"[^\"]+\\\\.(?:ad[ep]|ba[st]|chm|cmd|com|cpl|crt|eml|exe
|hlp|hta|in[fs]|isp|jse?|lnk|md[be]|ms[cipt]|pcd|pif|reg|scr|sct|shs|url|
vb[se]|ws[fhc])\")"
then
  fail text "This message has been rejected because it has\n\
             potentially executable content $1\n\
             This form of attachment has been used by\n\
             recent viruses or other malware.\n\
             If you meant to send this file then please\n\
             package it up as a zip file and resend it."
  seen finish
endif

# Same again using unquoted filename [content_type_unquoted_fn_match]
#
if $header_content-type: matches
"(?:file)?name=(\\\\S+\\\\.(?:ad[ep]|ba[st]|chm|cmd|com|cpl|crt|eml|exe|h
lp|hta|in[fs]|isp|jse?|lnk|md[be]|ms[cipt]|pcd|pif|reg|scr|sct|shs|url|vb
[se]|ws[fhc]))"
then
  fail text "This message has been rejected because it has\n\
             potentially executable content $1\n\
             This form of attachment has been used by\n\
             recent viruses or other malware.\n\
             If you meant to send this file then please\n\
             package it up as a zip file and resend it."
  seen finish
endif

## ----------------------------------------------------------------------
# Attempt to catch embedded VBS attachments in emails. These were
# used as the basis for the ILOVEYOU virus and its variants - many
# many varients. Quoted filename - [body_quoted_fn_match].
#
if $message_body matches "(?:Content-(?:Type:(?>\\\\s*)[\\\\w-]+/[\\\\w-
]+|Disposition:(?>\\\\s*)attachment);(?>\\\\s*)(?:file)?name=|begin(?>\\\
\s+)[0-
7]{3,4}(?>\\\\s+))(\"[^\"]+\\\\.(?:ad[ep]|ba[st]|chm|cmd|com|cpl|crt|eml|
exe|hlp|hta|in[fs]|isp|jse?|lnk|md[be]|ms[cipt]|pcd|pif|reg|scr|sct|shs|u
rl|vb[se]|ws[fhc])\")[\\\\s;]"
then
  fail text "This message has been rejected because it has\n\
             a potentially executable attachment $1\n\
             This form of attachment has been used by\n\
             recent viruses or other malware.\n\
             If you meant to send this file then please\n\
             package it up as a zip file and resend it."
  seen finish
endif

# Same again using unquoted filename [body_unquoted_fn_match].
if $message_body matches "(?:Content-(?:Type:(?>\\\\s*)[\\\\w-]+/[\\\\w-
]+|Disposition:(?>\\\\s*)attachment);(?>\\\\s*)(?:file)?name=|begin(?>\\\
\s+)[0-
7]{3,4}(?>\\\\s+))(\\\\S+\\\\.(?:ad[ep]|ba[st]|chm|cmd|com|cpl|crt|eml|ex
```

```
e|hlp|hta|in[fs]|isp|jse?|lnk|md[be]|ms[cipt]|pcd|pif|reg|scr|sct|shs|url
|vb[se]|ws[fhc]))[\\\\s;]"
then
  fail text "This message has been rejected because it has\n\
             a potentially executable attachment $1\n\
             This form of attachment has been used by\n\
             recent viruses or other malware.\n\
             If you meant to send this file then please\n\
             package it up as a zip file and resend it."
  seen finish
endif
```

NOTE: The above `system-filter` file can also be retrieved from the following URL: http://www.openna.com/products/books/securing-optimizing-linux/3rdedition/index.htm

Step2
Now, set the permission mode of the `system-filter` file to be (`0640/-rw-r-----`) and owned by the super-user '`root`' with group permission set to "`mail`" user for security reasons.

- To change the permission mode and ownership of `system-filter` file, use:
 `[root@deep /]#` **`chmod 640 /etc/mail/system-filter`**
 `[root@deep /]#` **`chown 0.12 /etc/mail/system-filter`**

`/etc/sysconfig/exim`: The `Exim` System Configuration File

The `/etc/sysconfig/exim` file is used to specify `EXIM` system configuration information, such as if `Exim` should run as a daemon, if it should listen for mail or not, and how much time to wait before sending a warning if messages in the queue directory have not been delivered.

Step 1
By default, the `exim` file does not exist after the installation, we have to create it.

- Create the **`exim`** file (`touch /etc/sysconfig/exim`) and add the lines:

```
# Run Exim as a daemon on the system. Remove the "-bd" option
# to run Exim as a Null Client Mail Server.
DAEMON="-bd"

# Procede the queue every 1 minutes.
QUEUE="-q1m"
```

The "`DAEMON=-bd`" option instructs `Exim` to run as a daemon. This line is useful when `Exim` client machines are configured to not accept mail directly from the outside in favor of forwarding all local mail to a Central Hub; not running a daemon also improves security. If you have configured your server or client machines in this way, all you have to do is to replace the `DAEMON="-bd"` option to `DAEMON=""`.

From time to time mail should be placed in a queue because it couldn't be transmitted immediately. The `QUEUE="-q1m"` sets the time interval before `Exim` retry to send messages again.

Step2
Now, set the permission mode of the `exim` file to be (`0644/-rw-r--r--`) and owned by the super-user '`root`' for security reason.

- To change the permission mode and ownership of `exim` file, use:
```
[root@deep /]# chmod 644 /etc/sysconfig/exim
[root@deep /]# chown 0.0 /etc/sysconfig/exim
```

`/etc/init.d/exim`: The `Exim` Initialization File

The `/etc/init.d/exim` script file is responsible for automatically starting and stopping the `Exim SMTP` server. Loading the `exim` daemon as a standalone daemon will eliminate load time and will even reduce swapping since non-library code will be shared.

Please note that the following script is only suitable for Linux operating systems that use `SystemV`. If your Linux system uses some other method, like `BSD`, you'll have to adjust the script below to make it work for you.

Step 1
Create the **`exim`** script file (`touch /etc/init.d/exim`) and add the following lines:

```
#!/bin/bash

# This shell script takes care of starting and stopping Exim.
#
# chkconfig: 2345 80 30
# description: Exim is a Mail Transport Agent, which is the program \
#              that moves mail from one machine to another.
#
# processname: exim
# config: /etc/mail/exim.conf
# pidfile: /var/run/exim.pid

# Source function library.
. /etc/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Source for additional options if we have them.
if [ -f /etc/sysconfig/exim ] ; then
        . /etc/sysconfig/exim
fi

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

# If Exim is not available stop now.
[ -f /usr/sbin/exim ] || exit 0

# Path to the Exim binary.
exim=/usr/sbin/exim

RETVAL=0
prog="Exim"

start() {
        echo -n $"Starting $prog: "
        daemon $exim $DAEMON $QUEUE
        RETVAL=$?
```

```
        echo
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/exim
        return $RETVAL
}

stop() {
        echo -n $"Shutting down $prog: "
        killproc $exim
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/exim
        return $RETVAL
}

# See how we were called.
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  status)
        status $exim
        RETVAL=$?
        ;;
  restart)
        stop
        start
        RETVAL=$?
        ;;
  condrestart)
        if [ -f /var/lock/subsys/exim ]; then
                stop
                start
                RETVAL=$?
        fi
        ;;
  *)
        echo $"Usage: $0 {start|stop|status|restart|condrestart}"
        exit 1
esac
exit $RETVAL
```

Step 2
Once the /etc/init.d/exim script file has been created, it is important to make it executable, change its default permissions, create the necessary links and then start it. Making this file executable will allow the system to run it, changing its default permission to allow only the root user to change it for security reasons, and the creation of the symbolic links will let the process control initialization of Linux, which is in charge of starting all the normal and authorized processes that need to run at boot time on your system, start the program automatically for you at each system reboot.

- To make this script executable and to change its default permissions, use the commands:
```
[root@deep /]# chmod 700 /etc/init.d/exim
[root@deep /]# chown 0.0 /etc/init.d/exim
```

- To create the symbolic rc.d links for Exim, use the following commands:
```
[root@deep /]# chkconfig --add exim
[root@deep /]# chkconfig --level 2345 exim on
```

- To start Exim software manually, use the following command:
```
[root@deep /]# /etc/init.d/exim start
Starting Exim:            [OK]
```

Testing Exim

Once our mailer software is configured and started, we have to run some tests to make sure Exim is working correctly on our system. The tests should all complete successfully or you will eventually lose mail messages. The first is to check that the run time configuration file of Exim is syntactically valid and the others are to make some simple routing tests to be sure that we can send and receive mails locally or remotely.

Again all tests should complete successfully or you could have problems sending or receiving mail messages with your mail server. To be able to successfully make the tests, we have to be the super-user "root" and execute all tests on the terminal of the server.

Test 1 - Checking that the run time configuration file of Exim is syntactically valid:
In this test, we will check that the run time configuration file of Exim is syntactically valid and does not contain any errors.

- To check that configuration file of Exim is syntactically valid, use the following command:
```
[root@deep /]# /usr/sbin/exim -bV
Exim version 4.05 #1 built 13-May-2002 01:35:36
Copyright (c) University of Cambridge 2002
```

If there are any errors in the configuration file, Exim will output error messages. Otherwise it just outputs the version number and builds date.

Test 2 - Verifying that Exim recognizes a local mailbox:
In this test, we should verify that Exim can recognize a local mailbox on the system.

- To verify that Exim can recognize a local mailbox, use the following command:
```
[root@deep /]# /usr/sbin/exim -bt postmaster
sysadmin@smtp.domain.com
    <-- postmaster@smtp.domain.com
  router = localuser, transport = local_delivery
```

Test 3 - Verifying that `Exim` recognizes a remote email address:
In this test, we should verify that `Exim` can recognize a remote email address on the Internet.

- To verify that `Exim` can recognize a remote address, use the following command:

```
[root@deep /]# /usr/sbin/exim -bt myaccount@hotmail.com
myaccount@hotmail.com
  router = dnslookup, transport = remote_smtp
  host mx11.hotmail.com [64.4.49.199]   MX=5
  host mx13.hotmail.com [64.4.50.71]    MX=5
  host mx08.hotmail.com [64.4.49.7]     MX=5
  host mx10.hotmail.com [64.4.49.135]   MX=5
  host mx07.hotmail.com [64.4.42.7]     MX=5
  host mx07.hotmail.com [65.54.236.7]   MX=5
  host mx02.hotmail.com [64.4.55.135]   MX=5
  host mx04.hotmail.com [64.4.56.135]   MX=5
  host mx06.hotmail.com [64.4.55.7]     MX=5
  host mx01.hotmail.com [64.4.55.71]    MX=5
  host mx09.hotmail.com [64.4.49.71]    MX=5
  host mx14.hotmail.com [65.54.232.7]   MX=5
  host mx05.hotmail.com [65.54.254.145] MX=5
  host mx12.hotmail.com [64.4.50.7]     MX=5
  host mx15.hotmail.com [65.54.232.71]  MX=5
```

Test 4 - Getting `Exim` to deliver mail locally:
In this test, we should verify that `Exim` can deliver mail locally on the system. We do it by passing messages directly to `Exim` on the terminal, without going through an external user agent (`MUA`).

- To verify that `Exim` can deliver mail locally, use the following command:

```
[root@deep /]# /usr/sbin/exim -v postmaster@smtp.domain.com
From: sysadmin@smtp.domain.com
To: postmaster@smtp.domain.com
Subject: Testing Exim

This is a test message.
^D
LOG: MAIN
  <= root@smtp.domain.com U=root P=local S=338
[root@smtp mail]# LOG: MAIN
  => sysadmin <postmaster@smtp.domain.com> R=localuser T=local_delivery
LOG: MAIN
  Completed
^C
```

The "`-v`" option causes `Exim` to output some verification of what it is doing. In this case you should see copies of three log lines, one for the message's arrival, one for its delivery, and one containing "Completed".

Test 5 - Getting `Exim` to deliver mail remotely:
In this test, we should verify that `Exim` can deliver mail remotely on the Internet. Again, we do it by passing messages directly to `Exim` on the terminal, without going through an external user agent (`MUA`).

- To verify that `Exim` can deliver mail remotely, use the following command:

```
[root@deep /]# /usr/sbin/exim -v myaccount@hotmail.com
From: sysadmin@smtp.domain.com
To: myaccount@hotmail.com
Subject: Testing Exim

This is a test message.
^D
LOG: MAIN
  <= root@smtp.domain.com U=root P=local S=324
[root@smtp ]# Connecting to mx08.hotmail.com [64.4.49.7]:25 ... connected
  SMTP<< 220-HotMail (NO UCE) ESMTP server ready at Mon, 13 May 2002
21:23:32 -0700
         220 ESMTP spoken here
  SMTP>> EHLO smtp.domain.com
  SMTP<< 250-hotmail.com Hello
         250-8bitmime
         250 SIZE 1572864
  SMTP>> MAIL FROM:<sysadmin@smtp.domain.com> SIZE=1357
  SMTP<< 250 Requested mail action okay, completed
  SMTP>> RCPT TO:<myaccount@hotmail.com>
  SMTP<< 250 Requested mail action okay, completed
  SMTP>> DATA
  SMTP<< 354 Start mail input; end with <CRLF>.<CRLF>
  SMTP>> writing message and terminating "."
  SMTP<< 250 Requested mail action okay, completed
  SMTP>> QUIT
LOG: MAIN
  => myaccount@hotmail.com R=dnslookup T=remote_smtp H=mx08.hotmail.com
[64.4.49.7]
LOG: MAIN
  Completed
^C
```

If you encounter problems, look at `Exim`'s log files (`/var/log/maillog`) to see if there is any relevant information there. Also be sure that your networking setting is correct, that your hostname is working, that your `DNS` resolves, that your firewall allows `SMTP` packets to pass, and that your `FQDN` (**F**ully **Q**ualified **D**omain **N**ame) is available.

Allowing Users to authenticate with Exim before relaying

An open relay mail server is very dangerous and the preferred method for spammers to abuse your system. Exim is built by default with Anti-Relay feature enable. This means that you cannot use it directly to send mail to someone and must authenticate before Exim allows you to relay. Try to send a message with your preferred MUA to a friend and you'll see that delivery of your message will fail. Different methods of authentication exist with Exim and it is up to us to choose which one we want to use and enable to allow relay.

Some methods like POP-Before-SMTP already exist on the Internet for a few MTA software but required you to hack the source code of your mail software to work and this is not what I really like as a solution. Other methods exist like using SMTP_AUTH, which is the method that we will use here since it is compatible with all MUA's on the market.

> **NOTE:** For more information about POP-Before-SMTP, please see: http://whoson.sourceforge.net

Authentication before relaying means that the user must be authenticated when he/she logs to the server to get or send his/her mail messages. In all cases this is done by connecting to a POP or IMAP server. How the SMTP_AUTH authentication works is explained as follow:

1. User connects to his/her POP or IMAP account on the server to get/send mail.
2. User send mail through its POP or IMAP server, Exim ask for username & password.
3. The MUA of the user send the username & password to Exim.
4. Exim compares information with its file, which handle username & password of the user.
5. If username & password correspond, then Exim allow relaying through the mail server.
6. If username & password do not correspond, then Exim send an error message.

In our configuration, we will allow user to relay with SMTP_AUTH. MS Outlook and Netscape use this kind of authentication, to the best of my knowledge, but many other MUA's use them too. The good news here is that we don't need to install any external programs since Exim has native support for SMTP_AUTH.

We will store allowed username & password in a file called exim.auth, I know an SQL database will be more adequate here for ISP's but I cannot explain this procedure since this is beyond the scope of the book. Therefore, we will use the exim.auth file, but the procedures to store usernames & passwords in a SQL database or a file are the same and only the configuration lines added to the exim.conf file differ.

Necessary steps to integrate SMTP_AUTH with Exim:

Procedures to allow SMTP_AUTH to run with Exim are not difficult to accomplish since the majority of the hack will happen inside the exim.conf file.

Step 1
First, we have to include authenticator specifications in this default configuration file. Adding the following lines under the “Authentication Configuration” section of the exim.conf file does this.

Add the following lines at the END of the “**Authentication Configuration**” section.

- Edit **exim.conf** file (vi /etc/mail/exim.conf) and add the following lines at the end of the "Authentication Configuration" part as follow:

```
# AUTH PLAIN authentication method used by Netscape Messenger.
#
plain:
  driver = plaintext
  public_name = PLAIN
  server_condition = "${if and {{!eq{$2}{}}{!eq{$3}{}} \
        {crypteq{$3}{${extract{1}{:} \
        {${lookup{$2}lsearch{/etc/mail/exim.auth} \
        {$value}{*:*}}}}}}{1}{0}}"

# AUTH LOGIN authentication method used by Outlook Express.
#
login:
  driver = plaintext
  public_name = LOGIN
  server_prompts = "Username:: : Password::"
  server_condition = "${if and {{!eq{$1}{}}{!eq{$2}{}} \
        {crypteq{$2}{${extract{1}{:} \
        {${lookup{$1}lsearch{/etc/mail/exim.auth} \
        {$value}{*:*}}}}}}{1}{0}}"
```

Step 2
Second, we have to create the exim.auth file that which will handle all mail usernames & passwords. Since we use PAM on our Linux system, username & password are stored into the /etc/shadow file and not inside the /etc/passwd file. This means that we have to make a copy of the shadow file in our /etc/mail directory and name it exim.auth.

- This can be done with the following command.
 `[root@deep /]# cp /etc/shadow /etc/mail/exim.auth`

Step 3
Now, set the permission mode of the exim.auth file to be (0640/-rw-r-----) and owned by the super-user ‘root’ with group permission set to “mail” user for security reason.

- To change the permission mode and ownership of exim.auth file, use the commands:
 `[root@deep /]# chmod 640 /etc/mail/exim.auth`
 `[root@deep /]# chown root.mail /etc/mail/exim.auth`

Step 4
The /etc/shadow file handles all user accounts on the Linux server and it is very dangerous to simply make a copy of it to the /etc/mail/exim.auth file, because if the mail server is compromised in any way, crackers will have access to all user accounts and will be able to use some password cracking software to get users passwords. Therefore, we have to edit it and remove any lines relating to system accounts like "root", "bin" and users from which a mail account is not provided.

To recap, you have to edit the exim.auth file and ONLY keep inside this file the lines related to users who have mail account access on the server. Any other lines relating, for example, to "root", "nobody", etc should absolutely be removed.

- Edit **exim.auth** file (vi /etc/mail/exim.auth) and remove any user lines you don't want to provide mail access on the server.

```
root:$1$hPNf/K/A$jFjIeW4B7Qf4F.zv/X0/h.:11817:0:99999:7::: ← Remove
bin:*:11817:0:99999:7::: ← Remove
daemon:*:11817:0:99999:7::: ← Remove
sync:*:11817:0:99999:7::: ← Remove
nobody:*:11817:0:99999:7::: ← Remove
named:!:11817:::::: ← Remove
rpm:!:11817:::::: ← Remove
gmourani:$1$99D6.K61$p/j3DljDTBMan/ZiUJMzW1:11821:::::: ← Keep
mail:!:11822:::::: ← Remove
```

In the above example, we only keep the user "gmourani" inside the exim.auth file because "gmourani" is the only user allowed to have a mail account on the server.

WARNING: The /etc/mail/exim.auth file should be recreated and modified each time you add a new mail user account to the server. Yes, you will have to repeat the above steps each time you add a new mail user account on the server. This is the only problem with this method.

Step 5
Finally, we have to restart the Exim daemon for the changes to take effect.

- To restart Exim, use the following command:

```
[root@deep /]# /etc/init.d/exim restart
Shutting down Exim:                [OK]
Starting Exim:                     [OK]
```

WARNING: Your MUA must be configured to support SMTP_AUTH. From my experience, Netscape works out of the box with SMTP_AUTH and you don't need to configure it for this purpose. On the other hand, MS Outlook needs some configuration from your part, you should make sure that the option under your Outlook account named "**My server requires authentication**" is checked. See you MUA manual for more information about how to enable SMTP_AUTH.

Running Exim with SSL support

This section applies only if you want to run Exim through an SSL connection. To begin our implementation of SSL into Exim we will first create the necessary certificate keys and add the required SSL parameters to the exim.conf file. Running Exim with SSL support is not for everyone. Before we embark on this, we need to first decide whether it is beneficial for us to do so. Some pros and cons are, but certainly not limited to, the following:

Pros:

- ✓ Client and server of a SMTP connection can be identified.
- ✓ The transmission of e-mail between a client and server utilizing SSL cannot be read and retranslated into plaintext provided a sufficiently secure cipher suite has been negotiated.
- ✓ The plaintext of e-mail between a client and server utilizing SSL cannot be modified by someone, provided a sufficiently secure cipher suite has been negotiated.

Cons:

- ✓ It does not provide end-to-end encryption, since a user can doesn't usually control the whole transmission. This is in contrast to the use of SSL for HTTP: here the user's client (a WWW browser) connects directly to the server that provides the data. E-mail can be transferred via multiple hops of which the sender can control at most only the first.
- ✓ It does not provide message authentication, unless the e-mail has been sent directly from the client's (SSL-capable) MUA to the recipients MTA that must record the client's certificate. Even then the message might be faked during local delivery.

Creating the necessary Exim certificate keys:

Below we'll show you how to create a certificate or a self-signed certificate with your own CA certificate for Exim. The principle is exactly the same as for creating a certificate or a self-signed certificate for a Web Server. We'll assume that your own CA certificates have been already created, if this is not the case, please refer to the OpenSSL chapter for further information.

Step 1
Here, we have to create a new SMTP certificate for Exim. This SMTP certificate becomes our private key and doesn't need to be encrypted. This is required for an unattended startup of Exim; otherwise you will have to enter the pass phrase each time Exim is started.

- To create a certificate private key without a pass phrase, use the following command:

```
[root@deep /]# cd /usr/share/ssl
[root@deep ssl]# openssl genrsa -rand
random1:random2:random3:random4:random5 -out smtp.key 1024
22383 semi-random bytes loaded
Generating RSA private key, 1024 bit long modulus
.++++++
.................................++++++
e is 65537 (0x10001)
```

Step 2
Once the private key has been created, we must generate a **C**ertificate **S**igning **R**equest (CSR) with the servers RSA private key. The command below will prompt you for the X.509 attributes of your certificate. If you prefer to have your Certificate Signing Request (CSR) signed by a commercial **C**ertifying **A**uthority (CA) like Thawte or Verisign you need to post the CSR file that will be generated below into a web form, pay for the signing, and await the signed Certificate.

- To generate the CSR, use the following command:

```
[root@deep ssl]# openssl req -new -key smtp.key -out smtp.csr
Using configuration from /usr/share/ssl/openssl.cnf
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [CA]:
State or Province Name (full name) [Quebec]:
Locality Name (eg, city) [Montreal]:
Organization Name (eg, company) [OpenNA.com SMTP Server]:
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) [smtp.openna.com]:
Email Address [noc@openna.com]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

WARNING: Be sure that you've entered the FQDN (**F**ully **Q**ualified **D**omain **N**ame) of the SMTP Server when OpenSSL prompts you for the "**Common Name**".

Step 3
This step is needed only if you want to sign, as your own CA, the csr certificate key. Now we must sign the new certificate with our own certificate authority that we have already created for generation of the Web Server certificate under the OpenSSL chapter (ca.crt). If the self signed CA certificate doesn't exist, then refer to the chapter related to OpenSSL for more information about how to create it.

- To sign with our own CA, the csr certificate, use the following command:

```
[root@deep ssl]# /usr/share/ssl/misc/sign smtp.csr
CA signing: smtp.csr -> smtp.crt:
Using configuration from ca.config
Enter PEM pass phrase:
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName           :PRINTABLE:'CA'
stateOrProvinceName   :PRINTABLE:'Quebec'
localityName          :PRINTABLE:'Montreal'
organizationName      :PRINTABLE:'OpenNA.com SMTP Server'
commonName            :PRINTABLE:'smtp.openna.com'
emailAddress          :IA5STRING:'noc@openna.com'
Certificate is to be certified until Feb 21 11:36:12 2003 GMT (365 days)
```

```
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
CA verifying: smtp.crt <-> CA cert
smtp.crt: OK
```

WARNING: If you receive an error message saying that the `csr` certificate that you are trying to sign already exists, it is because the information you have entered during the generation of the certificate key is the same as another `csr`, which you have already created. In this case, you must at least, change one bit of information in the new certificate key you want to create before signing the certificate with your own `CA`.

Step 4
Next, we should create the "`certs`" directory under which we will put the certificates keys. This directory should be created under the `/etc/mail` directory where all `Exim` files reside.

By default, the `certs` directory does not exist, we have to create it.

- To create the **`certs`** directory, use the following command:
```
[root@deep /]# mkdir -p /etc/mail/certs
```

Step 5
Now, set the permission mode of the `certs` directory to be (`0700/drwx------`) and owned by the user '`mail`' to allow `Exim` to access and reads certificates inside it.

- To change the permission mode and ownership of the `certs` directory, use:
```
[root@deep /]# chmod 700 /etc/mail/certs/
[root@deep /]# chown mail.mail /etc/mail/certs/
```

Step 6
Finally, we must place the certificates files (`smtp.key` and `smtp.crt`) to the appropriate directories for `Exim` to be able to find them when it starts up.

- To place the certificates into the appropriate directory, use the following commands:
```
[root@deep ssl]# mv smtp.key /etc/mail/certs/
[root@deep ssl]# mv smtp.crt /etc/mail/certs/
[root@deep ssl]# chmod 400 /etc/mail/certs/smtp.key
[root@deep ssl]# chmod 400 /etc/mail/certs/smtp.crt
[root@deep ssl]# chown mail.mail /etc/mail/certs/smtp.key
[root@deep ssl]# chown mail.mail /etc/mail/certs/smtp.crt
[root@deep ssl]# rm -f smtp.csr
```

With the above commands, we move the "`smtp.key`" file to the `/etc/mail/certs` directory and the "`smtp.crt`" file to the `/etc/mail/certs` directory. After that we change the permissions of both certificates to be only readable by the super-user '`root`' for security reasons and remove the "`smtp.csr`" file from our system since it is no longer needed.

Adding the required SSL parameters to the exim.conf file:

Once the Exim certificates have been created and moved to the appropriate location, we must add some new options into the exim.conf file for Exim to be configured to run with SSL support on the server.

Step 1

Below we show you the options to add into your default exim.conf file that are required for Exim to run with SSL support. Text in bold is what we have added to the default Exim configuration file.

- Edit your **exim.conf** file (vi /etc/mail/exim.conf), and add the following options inside the file to enable SSL support with Exim.

```
######################################################################
#                    MAIN CONFIGURATION SETTINGS                     #
######################################################################

primary_hostname = dev.openna.com

acl_smtp_rcpt = check_recipient
acl_smtp_data = check_message
acl_smtp_auth = check_auth

domainlist local_domains = @ : lsearch;/etc/mail/localdomains
hostlist relay_hosts = lsearch;/etc/mail/relaydomains
hostlist auth_relay_hosts = *
hostlist auth_over_tls_hosts = *
hostlist tls_relay_hosts = *

log_selector =  \
        +all_parents \
        +received_sender \
        +received_recipients \
        +smtp_confirmation \
        +smtp_syntax_error

allow_domain_literals = false
never_users = root:daemon:bin:sync:named
host_lookup = *
trusted_users = mail
gecos_pattern = ^([^,:]*)
gecos_name = $1
freeze_tell = postmaster
auto_thaw = 1h
ignore_bounce_errors_after = 30m
timeout_frozen_after = 7d

received_header_text = "Received: \
        ${if def:sender_rcvhost {from ${sender_rcvhost}\n\t}\
        {${if def:sender_ident {from ${sender_ident} }}\
        ${if def:sender_helo_name {(helo=${sender_helo_name})\n\t}}}}\
        by ${primary_hostname} \
        ${if def:received_protocol {with ${received_protocol}}} \
        (Exim ${version_number} #${compile_number} (OpenNA Linux))\n\t\
        id ${message_id}\
        ${if def:received_for {\n\tfor <$received_for>}}"

system_filter = /etc/mail/system-filter
message_body_visible = 5000
message_size_limit = 10M
smtp_accept_max = 2048
```

```
smtp_connect_backlog = 256
queue_only
split_spool_directory
queue_run_max = 1
remote_max_parallel = 1
rfc1413_hosts = *
rfc1413_query_timeout = 0s

smtp_banner = "Welcome on our mail server!\n\
        This system does not accept Unsolicited \
        Commercial Email\nand will blacklist \
        offenders via our spam processor.\nHave a \
        nice day!\n\n${primary_hostname} ESMTP Exim \
        ${version_number} ${tod_full}"

tls_advertise_hosts = *
tls_certificate = /etc/mail/certs/smtp.crt
tls_privatekey = /etc/mail/certs/smtp.key

#############################################################################
#                       ACL CONFIGURATION                                   #
#         Specifies access control lists for incoming SMTP mail             #
#############################################################################

begin acl

check_recipient:
  accept  hosts = :

  deny    local_parts   = ^.*[@%!/|]

  deny    senders       = *@dbm;/etc/mail/access.db : \
                          dbm;/etc/mail/access.db

  require verify        = sender

  deny    message       = unrouteable address
          hosts         = !127.0.0.1/8:0.0.0.0/0
         !verify        = recipient

  accept  domains       = +local_domains
          endpass
          message       = unknown user
          verify        = recipient

  accept  hosts         = +relay_hosts

  accept  hosts         = +auth_relay_hosts
          endpass
          message       = authentication required
          authenticated = *

  accept  hosts = +tls_relay_hosts
          endpass
          message = encryption required
          encrypted = *

  deny    message       = relay not permitted


check_message:
  accept
```

```
check_auth:
  accept  hosts = +auth_over_tls_hosts
          endpass
          message = STARTTLS required before AUTH
          encrypted = *
  accept

###############################################################################
#                      ROUTERS CONFIGURATION                                  #
#               Specifies how addresses are handled                           #
###############################################################################
#     THE ORDER IN WHICH THE ROUTERS ARE DEFINED IS IMPORTANT!                #
# An address is passed to each router in turn until it is accepted.           #
###############################################################################

begin routers

dnslookup:
  driver = dnslookup
  domains = ! +local_domains
  transport = remote_smtp
  ignore_target_hosts = 0.0.0.0 : 127.0.0.0/8
  no_more

system_aliases:
  driver = redirect
  allow_fail
  allow_defer
  data = ${lookup{$local_part}lsearch{/etc/mail/aliases}}
  user = mail
  file_transport = address_file
  pipe_transport = address_pipe

userforward:
  driver = redirect
  check_local_user
  file = $home/.forward
  no_verify
  no_expn
  check_ancestor
  allow_filter
  modemask = 002
  file_transport = address_file
  pipe_transport = address_pipe
  reply_transport = address_reply

localuser:
  driver = accept
  check_local_user
  transport = local_delivery

###############################################################################
#                      TRANSPORTS CONFIGURATION                               #
###############################################################################
#                       ORDER DOES NOT MATTER                                 #
#     Only one appropriate transport is called for each delivery.             #
###############################################################################

begin transports

remote_smtp:
  driver = smtp
```

```
local_delivery:
  driver = appendfile
  file = /var/mail/$local_part
  delivery_date_add
  envelope_to_add
  return_path_add
  group = mail
  mode = 0600

address_pipe:
  driver = pipe
  return_output

address_file:
  driver = appendfile
  delivery_date_add
  envelope_to_add
  return_path_add

address_reply:
  driver = autoreply

#############################################################################
#                       RETRY CONFIGURATION                                 #
#############################################################################

begin retry

# Domain               Error       Retries
# ------               -----       -------

*                      *           F,2h,15m; G,16h,1h,1.5; F,4d,6h

#############################################################################
#                       REWRITE CONFIGURATION                               #
#############################################################################

begin rewrite

#############################################################################
#                     AUTHENTICATION CONFIGURATION                          #
#############################################################################

begin authenticators

# AUTH PLAIN authentication method used by Netscape Messenger.
#
plain:
  driver = plaintext
  public_name = PLAIN
  server_condition = "${if and {{!eq{$2}{}}{!eq{$3}{}} \
        {crypteq{$3}{${extract{1}{:} \
        {${lookup{$2}lsearch{/etc/mail/exim.auth} \
        {$value}{*:*}}}}}}{1}{0}}"

# AUTH LOGIN authentication method used by Outlook Express.
#
login:
  driver = plaintext
  public_name = LOGIN
  server_prompts = "Username:: : Password::"
  server_condition = "${if and {{!eq{$1}{}}{!eq{$2}{}} \
```

```
{crypteq{$2}{${extract{1}{:} \
{${lookup{$1}lsearch{/etc/mail/exim.auth} \
{$value}{*:*}}}}}}{1}{0}}"
```

NOTE: Don't forget to restart your SMTP server for the changes to take effect.

```
[root@deep /]# /etc/init.d/exim restart
Shutting down Exim:               [OK]
Starting Exim:                    [OK]
```

Running Exim with Virtual Hosts support

This section applies only if you want to run Exim with virtual hosts support. Virtual hosts, or if you prefer virtual mail, is when you provide hosting on the same server for many domains and want to use Exim as the mail server for all the virtual domains hosted on the server. This is very interesting for hosting companies since they don't need to dedicate servers for virtual mail or even run more than one mail server to provide a mail service for all their virtual domains.

Adding the required Virtual Hosts parameters to the exim.conf file:

First off, we must add some new options into the exim.conf file for Exim to be configured to run with Virtual Hosts support. We have to be careful of the order in which the additional configuration lines related to Virtual Host are added to the exim.conf file.

Step 1
We have to include new router conditions for Exim to manage Virtual Domains. This is done by adding the following lines into the "Routers Configuration" section of the exim.conf file, after the "dnslookup" but before the "system_aliases" definition.

Add the following lines into the "**Routers Configuration**" part. Text in bold is what we have added to the default exim.conf file.

- Edit **exim.conf** file (vi /etc/mail/exim.conf) and add the following lines between "dnslookup" and "system_aliases" definitions as follow:

```
begin routers

dnslookup:
  driver = dnslookup
  domains = ! +local_domains
  transport = remote_smtp
  ignore_target_hosts = 0.0.0.0 : 127.0.0.0/8
  no_more

virtual_domains:
  driver = redirect
  allow_defer
  allow_fail
  data =
${expand:${lookup{$local_part@$domain}dbm*@{/etc/mail/virtualdomains.db}}
}
  retry_use_local_part

system_aliases:
  driver = redirect
  allow_fail
```

```
allow_defer
data = ${lookup{$local_part}lsearch{/etc/mail/aliases}}
user = mail
file_transport = address_file
pipe_transport = address_pipe
```

Step 2
Now, we have to restart the Exim daemon for the changes to take effect.

- To restart Exim, use the following command:

```
[root@deep /]# /etc/init.d/exim restart
Shutting down Exim:              [OK]
Starting Exim:                   [OK]
```

NOTE: For virtual domain to work, you must ensure that the MX record has been set on the primary and secondary DNS servers for the virtual domain. If the MX record doesn't exist, then set it up in your DNS servers before configuring virtual hosts. This is true for all mail server software.

Creating the necessary `virtualdomains` files:

The /etc/mail/virtualdomains file is used to define virtual aliasing mail accounts to virtual domains hosted on your server. You should use it every time you need to define aliases for virtual mail accounts on your system.

Step 1
By default, the virtualdomains file does not exist, we have to create it.

- Create the **virtualdomains** file (touch /etc/mail/virtualdomains) and add:

```
# This file must contains any email aliases for your virtual hosts users.
# For example, you do:
#
# support@virtual1.com: mark
# support@virtual2.com: john
#
# N.B.: Exim must be restarted after this file is modified.
# /usr/sbin/exim_dbmbuild virtualdomains virtualdomains.db
#
support@virtual1.com: mark
support@virtual2.com: john
```

In the above example, we permit any email addressed to "support" on the "virtual1.com" domain to be redirected to user "mark" on this virtual domain and any email addressed to "support" on the "virtual2.com" domain to be redirected to user "john" on this virtual domain. The both domains reside on the same server.

Step2
Now, set the permissions of the virtualdomains file to be (0640/-rw-r-----) and owned by the super-user 'root' with group permission set to "mail" user for security reasons.

- To change the permission mode and ownership of virtualdomains file, use:

```
[root@deep /]# chmod 640 /etc/mail/virtualdomains
[root@deep /]# chown root.mail /etc/mail/virtualdomains
```

Step 3
For every incoming or outgoing virtual connection, Exim looks up the sender's email address in the "virtualdomains" file. Because Exim may have to search through thousands of virtual email addresses in the "virtualdomains" file, it's a good idea to create a copy of the file in a separate "db" database format file to significantly improve lookup times.

A small program called "exim_dbmbuild" comes with Exim to achieve this. We can use it directly from the console each time we want to build/rebuild the "virtualdomains" database or create a script file to automate the process. Below, we show you both methods.

- To manually build/rebuild the virtualdomains database, use the following commands:
```
[root@deep /]# cd /etc/mail/
[root@deep mail]# exim_dbmbuild virtualdomains virtualdomains.db
```

- To automate the build/rebuild of the virtualdomains database, create a script file called "newvirtualdomains" under the /usr/sbin directory.
```
[root@deep /]# cd /usr/sbin/
[root@deep sbin]# touch newvirtualdomains
[root@deep sbin]# chmod 510 newvirtualdomains
[root@deep sbin]# chown 0.0 newvirtualdomains
```

- Now edit the **newvirtualdomains** script (vi /usr/sbin/virtualdomains) and add:
```
#!/bin/sh
exim_dbmbuild /etc/mail/virtualdomains /etc/mail/virtualdomains.db
/bin/chown root.mail /etc/mail/virtualdomains
/bin/chmod 640 /etc/mail/virtualdomains
/bin/chown root.mail /etc/mail/virtualdomains.db
/bin/chmod 640 /etc/mail/virtualdomains.db
```

NOTE: With the above "newvirtualdomains" script, you only need to run the script for the "virtualdomains" database of Exim to be automatically rebuilt with proper permissions and ownership.

Allowing Virtual Hosts to relay:

Remember that Exim by default does not allow you to relay without proper authentication, this is also true for virtual domains too. Your have to be sure that virtual domains user accounts (usernames & passwords) have been added to the exim.auth file and that the virtual domain in question is added the relaydomains and localdomains files to be allowed to relay. This is very important or relaying will be denied.

Step 1
Add all virtual mail usernames & passwords into the exim.auth file. This can be done by copying the shadow file to the /etc/mail directory by renaming it exim.auth as described earlier in this chapter.

- This can be done with the following command.
```
[root@deep /]# cp /etc/shadow /etc/mail/exim.auth
```

Step2
Now, add the virtual domain to the `localdomains` file for relaying to work.

- Edit your **`localdomains`** file (`vi /etc/mail/localdomains`) and add:

```
# localdomains - include all of your local domains name here.
# Virtual domains must be listed here to be recognized as local.
# N.B.: Exim must be restarted after this file is modified.
#
virtual1.com
```

Step 3
Finally, add the virtual domain to the `relaydomains` file for relaying to work.

- Edit your **`relaydomains`** file (`vi /etc/mail/relaydomains`) and add:

```
# This file handle all domains from which relaying is allowed.
# By default we include the locahost of the server or nothing will work.
# Virtual Domains must be added to this list or relaying will be denied.
# N.B.: Exim must be restarted after this file is modified.
#
localhost
virtual1.com
```

Step 4
Now, we have to restart the `Exim` daemon for the changes to take effect.

- To restart `Exim`, use the following command:

```
[root@deep /]# /etc/init.d/exim restart
Shutting down Exim:              [OK]
Starting Exim:                   [OK]
```

Running `Exim` with `Maildir` support

This section applies only if you want to run `Exim` with `Maildir` support. `Maildir` is a directory-based mail storage format originally introduced in the `Qmail` mail server, and adopted as an alternative mail storage format by both `Exim` and `Postfix`. The primary advantage of maildirs is that multiple applications can access the same `Maildir` simultaneously without requiring any kind of locking whatsoever. It is faster than native UNIX `mbox` format and a more efficient way to store mail, especially for `ISP's` with thousand of mail users.

One problem exists with this kind of setup, `Maildir` format is not supported by all `POP` or `IMAP` servers and remember that it's the `POP`/`IMAP` server that allow mail users to access their mail accounts to retrieve/send mails. Therefore, we should use a `POP`/`IMAP` server capable of providing `Maildir` format or this will not work. Fortunately, `tpop3d`, which is described later in this book, supports `Maildir` and we must use it with `Exim` if we want to run `Maildir`. Do not use `Qpopper` because it is not capable of providing the `Maildir` format.

Adding the required `Maildir` parameters to the `exim.conf` file:

First off, we must add some new options to the `exim.conf` file for `Exim` to be configured to run with `Maildir` support. We have to careful of the order in which additional configuration lines related to `Maildir` should be added to `exim.conf` file.

Step 1
To make `Maildir` to works with `Exim`, we have to change the default "`local_delivery`" options available under the "`Transports Configuration`" section of the `exim.conf` file.

Change the following lines in the "**Transports Configuration**" section. Text in bold is what has changed in the default `exim.conf` file.

- Edit **`exim.conf`** file (`vi /etc/mail/exim.conf`) and change the following lines:

```
local_delivery:
  driver = appendfile
  file = /var/mail/$local_part
  delivery_date_add
  envelope_to_add
  return_path_add
  group = mail
  mode = 0600
```

To read:

```
local_delivery:
  driver = appendfile
  check_string = ""
  create_directory
  delivery_date_add
  directory = ${home}/Maildir/
  directory_mode = 700
  envelope_to_add
  group = mail
  maildir_format
  message_prefix = ""
  message_suffix = ""
  mode = 0600
  return_path_add
```

Step 2
Now, we have to restart the `Exim` daemon for the changes to take effect.

- To restart `Exim`, use the following command:

```
[root@deep /]# /etc/init.d/exim restart
Shutting down Exim:              [OK]
Starting Exim:                   [OK]
```

Creating the necessary Maildir directory:

Once Exim has been restarted with the new configuration options, we have to create the Maildir directory under EACH user directory because Exim does not do it for us automatically.

Step 1
By default, the Maildir directory is not created by the mailer software, we have to do it under each user home directory deliver mails without error.

- To create the **Maildir** directory, use the following command:
 `[root@deep /]# mkdir -p /home/sysadmin/Maildir`

In the above example, I created a new Maildir directory for user "sysadmin". Don't forget to do it for each additional user on your system or they will not be able to receive mails.

Step2
Now, set the permissions of the Maildir directory to be (0700/drwx------) and owned by the user "sysadmin" in our case that owns the home directory.

- This can be done with the following command:
 `[root@deep /]# chown sysadmin.sysadmin /home/sysadmin/Maildir`

Running Exim with mail quota support

This section applies only if you want to run Exim with mail quota support. Yes, Exim has native support for mail quota, this simply means that with this feature, you don't need to use external program like "quota" anymore to provide quota support for your mail users and can use the Exim mail quota capability to archive the same result.

Before going into the steps to enable mail quota with Exim, it is important to note that the Maildir format as described earlier should be already enables with Exim. This is very important because mail quota is enabled on the user's home directory and if you don't use the Maildir feature of Exim, then you'll be unable to take advantage of the Exim mail quota support. That said, now lets go to the required steps.

Adding the required Quota parameters to the exim.conf file:

We must add some new options to the exim.conf file for Exim to be configured to run with Quota support. Again, we have to be careful of the order in which additional configuration lines related to Quota should be added to the exim.conf file.

Step 1
To make Quota work with Exim, we have to change the default "local_delivery" options available under the "Transports Configuration" section of the exim.conf file. Here we assume that Maildir support is already set in your Exim configuration file as shown below.

Change the following lines into the "**Transports Configuration**" part. Text in bold is what we have changed to the default exim.conf file.

- Edit **exim.conf** file (vi /etc/mail/exim.conf) and change the following lines:

```
local_delivery:
  driver = appendfile
  check_string = ""
  create_directory
```

```
  delivery_date_add
  directory = ${home}/Maildir/
  directory_mode = 700
  envelope_to_add
  group = mail
  maildir_format
  message_prefix = ""
  message_suffix = ""
  mode = 0600
  return_path_add
```

To read:

```
local_delivery:
  driver = appendfile
  check_string = ""
  create_directory
  delivery_date_add
  directory = ${home}/Maildir/
  directory_mode = 700
  envelope_to_add
  group = mail
  maildir_format
  maildir_tag = ,S=$message_size
  message_prefix = ""
  message_suffix = ""
  mode = 0600
  quota = 10M
  quota_size_regex = S=(\d+)$
  quota_warn_threshold = 75%
  return_path_add
```

Step 2
Now, we have to restart `Exim` daemon for the changes to take effect.

- To restart `Exim`, use the following command:

```
[root@deep /]# /etc/init.d/exim restart
Shutting down Exim:              [OK]
Starting Exim:                   [OK]
```

Running `Exim` as a Null Client Mail Server

Now we have explained how to configure `Exim` to send and receive mail to and from both the internal and the external network. This kind of configuration is also known as a Central Mail Hub Server and it useful when you need to have a fully operational mail server. In some cases we simply don't need this kind of complete setup because the server on which we want to run `Exim` is not configured to be principally a mail server, but to run other services like web, ftp, dns, gateway, etc. The only mail generated on this type of server should be mail dedicated to system account.

For these servers, we can configure `Exim` to run as a Null Client Mail Server. A Null Client Mail Server is a mail server that doesn't listen for incoming connections on the server to receive mail. It can only send all locally generated mails dedicated to local system account to where you want them to go. This means that a Null Client Mail Server can only deliver locally generated mails to a Central Mail Hub Server of your choice.

This is useful to considerably improve the security of your mail server because it does not listen for incoming mail connections. Spammers also cannot use it to forward mail to some place on the Internet. Here is an overview of both possible configurations for a mail server:

A Central Mail Hub Server configuration:
Instead of having each individual server or workstation in a network handle its own mail, it can be advantageous to have powerful central server that handles all mail. Such a server is called a Central Mail Hub Server. The advantage of a Central Mail Hub Server is:

- ✓ Can receive or send mail to everywhere.
- ✓ Allow mail users to connect to their mail account to send or get mail.
- ✓ Capable to receive and manage mail coming from null client mail servers.

A Null Client Mail Server configuration:
A mail service is indispensable for all types of server, even if the task of the server in question is not to process mail, because we should at least be able to get important messages generated locally and addressed to the local system account (`posmaster`) on the server.

A null client mail server never receives mail directly from the outside world, and relays (sends) all their mail through the Central Mail Hub Server. In this way, important messages addressed to the local system account can be delivered remotely to the Central Mail Hub Server for verification. The advantage of a Null Client Mail Server is:

- ✓ No mail is sent directly to a local user account on the system.
- ✓ All mails are sent (forwarded) to the Central Mail Hub Server.
- ✓ No client's name needs to be known to the outside world.
- ✓ `Exim` daemon does not need to listen for incoming mail connections.

Making `Exim` to not listen for incoming mail connection:

When `Exim` is configured to run as a Null Client Mail Server, it should not listen for incoming connections on your server. This is very important for security reasons and we must edit our `/etc/sysconfig/exim` file to make the changes.

Step 1
For `Exim` to run as a daemon on the system it should be configured with the "`-bd`" option. This is what we do by default when we install the software. We must change this when `Exim` is running as a Null Client Mail Server.

- Edit the **`exim`** file (`vi /etc/sysconfig/exim`) and change the following line:

  ```
  DAEMON="-bd"
  ```

 To read:

  ```
  DAEMON=""
  ```

Setting "postmaster" to a remote address into the aliases file:

Since local users on the server cannot receive mail anymore, we must change the "postmaster" alias in the /etc/mail/aliases file to the address of the remote system administrator on the Mail Hub Server that should receive all generated local mail.

Step 1

In the following example, I use the aliases file that we have created previously in this chapter and change the "postmaster" alias definition for a human email address, on the Central Mail Hub Sever, that should now receive all local mails generated on the Null Client Mail Server. Text in bold is what I've changed from the original example aliases file.

- Edit the **aliases** file (vi /etc/mail/aliases) and change the following line:

```
# The following aliases are required by the mail RFCs 2821 and 2822.
# At least, you should set "postmaster" to the address of a HUMAN
# who deals with this system's mail problems.
#
postmaster:     sysadmin@domain.com
mailer-daemon:  postmaster
root:           postmaster

# It is a good idea to redirect any messages sent to system accounts
# so that they don't just get ignored.
#
bin:            root
daemon:         root
sync:           root
mail:           root
pop:            root
uucp:           root
ftp:            root
nobody:         root
www:            root
named:          root
postgres:       root
mysql:          root
squid:          root
amavis:         root
operator:       root
abuse:          root
hostmaster:     root
webmaster:      root
```

To read:

```
# The following aliases are required by the mail RFCs 2821 and 2822.
# At least, you should set "postmaster" to the address of a HUMAN
# who deals with this system's mail problems.
#
postmaster:     markus@mailhubserver.com
mailer-daemon:  postmaster
root:           postmaster

# It is a good idea to redirect any messages sent to system accounts
# so that they don't just get ignored.
#
bin:            root
daemon:         root
sync:           root
mail:           root
pop:            root
```

```
uucp:           root
ftp:            root
nobody:         root
www:            root
named:          root
postgres:       root
mysql:          root
squid:          root
amavis:         root
operator:       root
abuse:          root
hostmaster:     root
webmaster:      root
```

Step 2
Now, we have to restart the `Exim` daemon for the changes to take effect.

- To restart `Exim`, use the following command:
```
[root@deep /]# /etc/init.d/exim restart
Shutting down Exim:             [OK]
Starting Exim:                  [OK]
```

`Exim` Administrative Tools

The commands listed below are some that we use often, but many more exist. Check the manual page and documentation for more information.

`newaliases`

The purpose of the "`newaliases`" script utility of `Exim`, which we have created previously in this chapter, is to rebuild and update the database for the `aliases` file `/etc/mail/aliases`. It must be run each time you change the contents of this file in order for the changes to take effect.

- To update the `aliases` file with the "`newaliases`" utility, use the following command:
```
[root@deep /]# /usr/sbin/newaliases
21 entries written
```

`newaccess`

The purpose of the "`newaccess`" script utility of `Exim`, which we have created previously in this chapter, is to rebuild and update the database for the `access` file `/etc/mail/access`. It must be run each time you change the contents of this file in order for the changes to take effect.

- To update the `access` file with the "`newaccess`" utility, use the following command:
```
[root@deep /]# /usr/sbin/newaccess
13 entries written
```

`newvirtualdomains`

The purpose of the "`newvirtualdomains`" script utility of `Exim` is to rebuild and update the database for the `virtualdomains` file `/etc/mail/virtualdomains`. It must be run each time you change the contents of this file in order for the changes to take effect.

- To update the `virtualdomains` file with the "`newvirtualdomains`" utility, use:
```
[root@deep /]# /usr/sbin/newvirtualdomains
20 entries written
```

mailq

The purpose of the "mailq" program utility of Exim is to print a summary of the mail messages queued waiting for future delivery. This could happen for different reasons.

- To print a summary of the mail messages queued, use the following command:
 [root@deep /]# **/usr/bin/mailq**

- To process all messages in the queue manually, use the following command:
 [root@deep /]# **/usr/sbin/exim -qf**

frozen messages

A frozen message happens when Exim cannot deliver the mail to its destination because the recipient does not exist, because the mail is fake or because the "from" header of the message is empty and so on. Many reasons exist for which a frozen message could happen and generally the reason is a good reason.

If the message is frozen, attempts to deliver it are suspended. Frozen messages are resubmitted after a period of time, as defined into the exim.conf file (in our case each hour). If the mailer cannot deliver the message, then it is refrozen for one additional hour before delivery takes place again. If after one week, the message is still frozen in the queue, then Exim will remove it.

This could be ok in most cases, but with spammers using fake email addresses when they try to pass some spam to your mail server, this could pose problems since your queue will become too big. To solve the problem we have some solutions, here I show you some interesting options that you could use.

- To manually unfreeze frozen messages in the queue, use the following command:
 [root@deep /]# **/usr/sbin/exim -Mt <message id> <message id>**

Where you replace <message id> by the actual identifier for a queued message. The above option requests Exim to "thaw" any of the listed messages that are "frozen", so that delivery attempts can resume.

- To see the content of the message body in the queue, use the following command:
 [root@deep /]# **/usr/sbin/exim -Mvb <message id>**

Where you replace <message id> by the actual identifier for a queued message. The above option causes the content of the message body (-D) spool file to be written to the standard output.

- To manually remove messages in the queue, use the following command:
 [root@deep /]# **/usr/sbin/exim -Mrm <message id> <message id>**

Where you replace <message id> by the actual identifier for a queued message. The above option requests Exim to remove the given messages from the queue. No bounce messages are sent; each message is simply forgotten.

Further documentation

For more details about Exim program, there is one manual page that you should read:

$ man exim (8) - Mail Transfer Agent configuration.

CHAPTER 31

Qmail

IN THIS CHAPTER

1. **Compiling, Optimizing & Installing `Qmail`**
2. **Configuring `Qmail`**
3. **Testing `Qmail`**
4. **Allowing Users to authenticate with `Qmail` before relaying**
5. **Running `Qmail` with `SSL` support**
6. **Running `Qmail` with Virtual Hosts support**
7. **Running `Qmail` as a Null Client Mail Server**
8. **Running `Qmail` as a Mini-Qmail Mail Server**
9. **Running `qmail-pop3d` with `SSL` support**
10. **`Qmail` Administrative Tools**
11. **`Qmail` Users Tools**

Linux Qmail

Abstract

If you decide to use Qmail successfully as a Mail Server, you must be aware of how it works. It is completely different to Exim. The Qmail system is built using the philosophy of having many small utilities that do one thing, and then combining these utilities to make something useful happen. Qmail delivery takes place using a number of separate programs that communicate with each other in well-defined ways.

Finally, and before going into Qmail deeper, it's important to note that Qmail runs through a program named tcpserver; which functions in the much the same manner as Xinetd, but faster. Personally, I think that there are too many add-ons for Qmail to be able to run it. On the other hand, if we look at some surveys, we'll find that Hotmail, with thirty million plus users, has been using Qmail for outgoing mail since 1997. (Reportedly, after Microsoft purchased Hotmail, it tried to move Hotmail to Microsoft Exchange under Windows NT. Exchange crashed.)

Qmail is a secure, reliable, efficient, simple message transfer agent. It is meant as a complete replacement for the entire sendmail-binmail system on typical Internet-connected UNIX hosts. Security isn't just a goal, but an absolute requirement. Mail delivery is critical for users; it cannot be turned off, so it must be completely secure.

Qmail supports host and user masquerading, full host hiding, virtual domains, null clients, list-owner rewriting, relay control, double-bounce recording, arbitrary RFC 822 address lists, cross-host mailing list loop detection, per-recipient checkpointing, downed host backoffs, independent message retry schedules, etc. In short, it's up to speed on modern MTA features. Qmail also includes a drop-in "sendmail" wrapper so that it will be used transparently by your current UAs.

With Qmail only one Qmail program is setuid: qmail-queue. Its only purpose is to add a new mail message to the outgoing queue. Also five of the most important Qmail programs are not security-critical. Even if all of these programs are completely compromised, so that an intruder has full control over the program accounts and the mail queue, he still can't take over your system. Finally, the stralloc concept and getln() of qmail which comes from a basic C library make it very easy to avoid buffer overruns, memory leaks, and artificial line length limits.

As with the previous Exim set up, we'll show you two different configurations that you can use for Qmail; one for a Central Mail Hub Server, and another for a Null Client Mail Server, which can be used for any server that doesn't run as a Mail Hub Server.

Finally, I'd like to advise you that external programs like logcheck, tripwire, etc do not support Qmail. It can be very difficult to make it work with these kinds of programs and trying to find help on the Qmail mailing list can also be very difficult, since support is not as you would expect it to be, like with Exim. A lot of serious questions are asked without any answers and only stupid questions seem to be answered by the mailing list users (I'm sorry but it is true). Therefore, and before going into the compilation and installation of this software, I recommend you think about your decision.

As I've mentioned before, Qmail uses a modular design to build everything into a single binary. This means, for example, that its binary program, which is responsible for sending mail, is separate from its program that is responsible for receiving mails, and so on. In order to perform other useful actions, we will need to install some additional utilities in this chapter.

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest Qmail version number is 1.03

The procedures given in this chapter are likely to work on all Linux platforms, but we have only tested it on OpenNA Linux and Red Hat Linux.

Packages

The following is based on information listed by Qmail as of 2002/06/24. Please check regularly at http://www.qmail.org/top.html for the latest status. We chose to install the required component from source because it provides the facility to fine tune the installation.

Source code is available from:
Qmail Homepage: http://www.qmail.org/top.html
Qmail FTP Site: 131.193.178.181
You must be sure to download: qmail-1.03.tar.gz

Prerequisites

Qmail requires that the below software is already installed on your system to be able to compile successfully. If this is not the case, you must install it. Please make sure you have these programs installed on your machine before you proceed with this chapter.

- ✓ OpenSSL is required to run Qmail with SSL support on your system.
- ✓ UCSPI-TCP is needed by Qmail and should be already installed on your system.
- ✓ Checkpassword is needed by Qmail and should be already installed on your system.
- ✓ Fastforward is needed by Qmail and should be already installed on your system.

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all files installed on the system in the event of an update in the future. To solve the problem, it is a good idea to make a list of files on the system before you install Qmail, and one afterwards, and then compares them using the **diff** utility to find out what files were placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep root]# find /* > Qmail1
  ```
- And the following one after you install the software:
  ```
  [root@deep root]# find /* > Qmail2
  ```
- Then use the following command to get a list of what changed:
  ```
  [root@deep root]# diff Qmail1 Qmail2 > Qmail-Installed
  ```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. Related to our example above, we use the `/root` directory of the system to stock all generated list files.

Compiling, Optimizing & Installing `Qmail`

Below are the required steps that you must make to configure, compile and optimize the `Qmail` software before installing it into your Linux system. First off, we install the program as user "`root`" so as to avoid any permissioning problems.

As you'll see later, `Qmail` has no pre-compilation configuration; instead `Qmail` automatically adapts itself to your UNIX variant and allows a quick installation. On the other hand, due to its quick installation feature, it doesn't let us install different parts of the software where we want them to go and this is why we must do a bit of tweaking to make it fit our system environment.

Step 1
Once you get the program from the main software site you must copy it to the `/var/tmp` directory and change to this location before expanding the archive.

- This can be done with the following commands:

```
[root@deep /]# cp qmail-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf qmail-version.tar.gz
```

Step 2
In order to check that the version of `Qmail`, which you are going to install, is an original and unmodified one, use the command described below to check its `MD5` hashes checksum.

- To verify the `MD5` checksum of `Qmail`, use the following command:

```
[root@deep tmp]# md5sum qmail-1.03.tar.gz
```

This should yield an output similar to this:

```
622f65f982e380dbe86e6574f3abcb7c  qmail-1.03.tar.gz
```

Now check that this checksum is exactly the same as the one published on the `Qmail` website at the following URL: http://cr.yp.to/qmail/dist.html

Step 3
`Qmail` cannot run as super-user root; for this reason we must create special users and groups with no shell privileges on the system for running `Qmail` daemons. It's important to note that no `Qmail` users or groups have a shell account on the system; this is an important security point to consider.

During the creation of all the required Qmail accounts as shown below, we'll redirect all Qmail users and groups account to a /bin/false shell. Once again this is an important security measure to take.

- To create these special Qmail users on OpenNA Linux, use the following commands:

```
[root@deep tmp]# groupadd -g 81 nofiles > /dev/null 2>&1 || :
[root@deep tmp]# groupadd -g 82 qmail > /dev/null 2>&1 || :

[root@deep tmp]# useradd -c "Mail Server" -d /var/qmail -g 81 -s
/bin/false -u 81 qmaild > /dev/null 2>&1 || :

[root@deep tmp]# useradd -c "Mail Server" -d /var/qmail/alias -g 81 -s
/bin/false -u 82 alias > /dev/null 2>&1 || :

[root@deep tmp]# useradd -c "Mail Server" -d /var/qmail -g 82 -s
/bin/false -u 83 qmailq > /dev/null 2>&1 || :

[root@deep tmp]# useradd -c "Mail Server" -d /var/qmail -g 82 -s
/bin/false -u 84 qmailr > /dev/null 2>&1 || :

[root@deep tmp]# useradd -c "Mail Server" -d /var/qmail -g 82 -s
/bin/false -u 85 qmails > /dev/null 2>&1 || :

[root@deep tmp]# useradd -c "Mail Server" -d /var/qmail -g 81 -s
/bin/false -u 86 qmaill > /dev/null 2>&1 || :

[root@deep tmp]# useradd -c "Mail Server" -d /var/qmail -g 81 -s
/bin/false -u 87 qmailp > /dev/null 2>&1 || :
```

- To create these special Qmail users on Red Hat Linux, use the following commands:

```
[root@deep tmp]# groupadd -f -g 81 nofiles > /dev/null 2>&1 || :
[root@deep tmp]# groupadd -f -g 82 qmail > /dev/null 2>&1 || :

[root@deep tmp]# useradd -c "Mail Server" -g 81 -u 81 -s /bin/false -r -d
/var/qmail qmaild > /dev/null 2>&1 || :

[root@deep tmp]# useradd -c "Mail Server" -g 81 -u 82 -s /bin/false -r -d
/var/qmail/alias alias > /dev/null 2>&1 || :

[root@deep tmp]# useradd -c "Mail Server" -g 82 -u 83 -s /bin/false -r -d
/var/qmail qmailq > /dev/null 2>&1 || :

[root@deep tmp]# useradd -c "Mail Server" -g 82 -u 84 -s /bin/false -r -d
/var/qmail qmailr > /dev/null 2>&1 || :

[root@deep tmp]# useradd -c "Mail Server" -g 82 -u 85 -s /bin/false -r -d
/var/qmail qmails > /dev/null 2>&1 || :

[root@deep tmp]# useradd -c "Mail Server" -g 81 -u 86 -s /bin/false -r -d
/var/qmail qmaill > /dev/null 2>&1 || :

[root@deep tmp]# useradd -c "Mail Server" -g 81 -u 87 -s /bin/false -r -
d /var/qmail qmailp > /dev/null 2>&1 || :
```

The above command will create all the required Qmail groups and users null accounts, with no passwords, no valid shells, no files owned; nothing but a UID and a GID for the program to run properly and in a secure manner. Remember that the Qmail daemon does not need to have shell accounts on the server.

Step 4
Now, edit the **shells** file (vi /etc/shells) and add a non-existent shell name "/bin/false", which is the one we used in the passwd command above.

```
[root@deep tmp]# vi /etc/shells
/bin/bash2
/bin/bash
/bin/sh
/bin/false ← This is our added no-existent shell
```

Step 5
After that, move into the newly created Qmail directory and create the qmail home directory manually. The qmail home directory is where everything related to Qmail software is stored. In our configuration, we will create some links pointing to the /etc and /usr/bin directories because we want to relocate Qmail files into this directory to be compliant with our operating system and to simplify the administration tasks of the mail server.

- To move into the newly created Qmail archive directory, use the following command:
```
[root@deep tmp]# cd qmail-1.03/
```

- To create the qmail home directory and requird links, use the following command:
```
[root@deep qmail-1.03]# mkdir /var/qmail
[root@deep qmail-1.03]# chown 0.qmail /var/qmail
[root@deep qmail-1.03]# mkdir -p /etc/qmail/alias
[root@deep qmail-1.03]# mkdir -p /etc/qmail/control
[root@deep qmail-1.03]# mkdir -p /etc/qmail/users
[root@deep qmail-1.03]# ln -sf /etc/qmail/alias /var/qmail
[root@deep qmail-1.03]# ln -sf /etc/qmail/control /var/qmail
[root@deep qmail-1.03]# ln -sf /etc/qmail/users /var/qmail
[root@deep qmail-1.03]# ln -sf /usr/bin /var/qmail/bin
[root@deep qmail-1.03]# ln -sf /usr/share/man /var/qmail/man
```

Step 6
Before going into the compilation of the program, we'll edit the conf-cc file and change the default compiler flags to fit our own CPU architecture for better performance.

- Edit the **conf-cc** file (vi conf-cc) and change the line:

  ```
  cc -O2
  ```

 To read:

  ```
  gcc -O2 -march=i686 -funroll-loops
  ```

Making Qmail to compile with SMTP_AUTH support:

There is an external patch available for Qmail that allows us to compile it with SMTP_AUTH support. If you are interested in compiling Qmail to support SMTP_AUTH for Anti-Relay protection, then I recommend you follow these steps. If you don't want to compile Qmail with SMTP_AUTH support, you can simply skip these steps and go directly to next section where we will compile the software for our system. I highly recommend you to compile Qmail with SMTP_AUTH support if you want to stop spammers using your server as an open relay.

Step 1
First of, we have to retrieve the SMTP_AUTH patch available on the Internet. This patch can be downloaded from the following location: http://members.elysium.pl/brush/qmail-smtpd-auth/

Step 2
Once you have a copy of this patch, you should move it under the /var/tmp directory and patch your Qmail source files.

- This can be done with the following commands:

```
[root@deep /]# mv qmail-smtpd-auth-0.31.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf qmail-smtpd-auth-0.31.tar.gz
[root@deep tmp]# cd qmail-smtpd-auth-0.31
[root@deep qmail-smtpd-auth-0.31]# cp auth.patch ../qmail-1.03/
[root@deep qmail-smtpd-auth-0.31]# cp base64.c ../qmail-1.03/
[root@deep qmail-smtpd-auth-0.31]# cp base64.h ../qmail-1.03/
[root@deep qmail-smtpd-auth-0.31]# cd ../qmail-1.03/
[root@deep qmail-1.03]# patch -p0 < auth.patch
```

Compiling Qmail:

Now, we must make a list of files on the system before installing the software, and one afterwards, and then compare them using the **diff** utility to find out what files are placed where and finally compile and install the Qmail software.

Step 1
Compile and install Qmail with the following commands.

```
[root@deep qmail-1.03]# make
[root@deep qmail-1.03]# make man
[root@deep qmail-1.03]# cd
[root@deep root]# find /* > Qmail1
[root@deep root]# cd /var/tmp/qmail-1.03/
[root@deep qmail-1.03]# make setup check
[root@deep qmail-1.03]# ln -s /var/qmail/bin/sendmail /usr/lib/sendmail
[root@deep qmail-1.03]# ln -s /var/qmail/bin/sendmail /usr/sbin/sendmail
[root@deep qmail-1.03]# ln -sf /var/qmail/bin/qmail-qread /usr/bin/mailq
[root@deep qmail-1.03]# rm -rf /var/qmail/boot/
[root@deep qmail-1.03]# rm -rf /var/qmail/doc/
[root@deep qmail-1.03]# maildirmake /etc/skel/Maildir
[root@deep qmail-1.03]# cd
[root@deep root]# find /* > Qmail2
[root@deep root]# diff qmail1 qmail2 > Qmail-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

Step 2
You MUST tell Qmail your hostname. To do this, use the config script of Qmail, which looks up your host name in DNS. This config script will also look up your local IP addresses in DNS to decide which hosts to it should accept mail for. By default is will only accept incoming mail connection for your default hostname. This is a security feature to avoid spam and open relay on the mail server.

```
[root@deep root]# cd /var/tmp/qmail-1.03/
[root@deep qmail-1.03]# ./config
Your hostname is smtp
Your host's fully qualified name in DNS is smtp.domain.com.
Putting smtp.domain.com into control/me...
Putting domain.com into control/defaultdomain...
Putting domain.com into control/plusdomain...

Checking local IP addresses:
127.0.0.1: Adding localhost to control/locals...
1.2.3.4: Adding smtp.domain.com to control/locals...

If there are any other domain names that point to you,
you will have to add them to /var/qmail/control/locals.
You don't have to worry about aliases, i.e., domains with CNAME records.

Copying /var/qmail/control/locals to /var/qmail/control/rcpthosts...
Now qmail will refuse to accept SMTP messages except to those hosts.
Make sure to change rcpthosts if you add hosts to locals or
virtualdomains!
```

NOTE: If you receive an error message like:

```
Your hostname is smtp.
hard error
Sorry, I couldn't find your host's canonical name in DNS.
You will have to set up control/me yourself.
```

You'll have to run the config-fast script located in the same source directory as follows:

```
./config-fast smtp.domain.com
```

Here I assume that your domain is domain.com and the hostname of your computer is smtp.

Step 3
Now it's time to add the minimum required aliases for Qmail to run properly on your system. You should set up at least aliases for Postmaster, Mailer-Daemon, and root. For security reasons the super-user "root" never receives mail with Qmail, this is the same as Exim. Because many programs on our server need to send system messages to "postmaster", "mailer-daemon" or "root", we can create an alias to another user locally or remotely.

Finally an important note is the fact that Qmail uses files for every alias. This is one of the major ways that Qmail differs from Exim. Therefore don't forget to create a ".qmail" alias file for every regular users on the system that need it and only if users need to have alias file. Users' alias file should be created under their home directory and named ".qmail".

- Create minimal aliases for accounts "postmaster", "mailer-daemon", and "root".

```
[root@deep qmail-1.03]# cd ~alias
[root@deep alias]# touch .qmail-postmaster
[root@deep alias]# touch .qmail-mailer-daemon
[root@deep alias]# touch .qmail-root
[root@deep alias]# echo sysadmin > .qmail-postmaster
[root@deep alias]# echo sysadmin > .qmail-mailer-daemon
[root@deep alias]# echo sysadmin > .qmail-root
[root@deep alias]# chmod 644 .qmail-*
[root@deep alias]# chown root.nofiles .qmail-*
```

Here, we instruct Qmail to send all message intended to "postmaster", "mailer-daemon" or the super-user "root" to a local non-privileged user account named sysadmin.

NOTE: Qmail doesn't have any built-in support for Sendmail /etc/aliases. If you have big /etc/aliases and you'd like to keep it, install the "fastforward" package, which is available separately from the Qmail website. This package "fastforward" is discussed later in this chapter. As a security precaution, Qmail refuses to deliver mail to users who don't own their home directory. In fact, such users aren't even considered users by Qmail. As a result, if "postmaster" doesn't own ~postmaster, then "postmaster" isn't a user, and postmaster@domain.com isn't a valid mailbox. This is why the above aliases are important to set.

Step 4
The Qmail package, once installed on your system, includes a local delivery agent, called 'qmail-local', which provides user-controlled mailing lists, cross-host alias loop detection, and many other important Qmail features, like the Qmail crashproof Maildir directory for your incoming mail messages. This Qmail program (qmail-local) is intended to replace binmail which is the default UNIX /bin/mail program used under Linux to delivers mail locally into a central spool directory called /var/spool/mail.

There's one important difference between qmail-local and binmail: qmail-local delivers mail by default into ~user/Mailbox or ~user/Maildir, rather than /var/spool/mail.

What does this imply?
There are two basic problems with /var/spool/mail:

- ✓ It's slow. On systems with thousands of users, /var/spool/mail has thousands of entries. A few UNIX systems support fast operations on large directories, but most don't.
- ✓ It's insecure. Writing code that works safely in a world-writeable directory is not easy. See, for example, CERT advisory 95:02.

For these reasons, and to tighten the security of our configured system, as well as to optimize the Qmail Mail Server to perform at its peak, we'll change and configure the mail software to look at the Qmail ~user/Maildir directly. Maildir is a feature of Qmail to replace the old well-known Unix Mailbox directory that is less reliable than Maildir.

Usually, all future users in the system will have the Maildir directory automatically created for them by Linux because we have added the required Maildir skeleton into the /etc/skel directory on our server. For all existing users in the system, you have to create this new Maildir directory manually as follows.

- To create a new Maildir for all existing users in the system, use the command:
  ```
  [root@deep /]# maildirmake $HOME/Maildir
  ```

Where <$HOME> is the username directory where you want to create this new Qmail Maildir directory for all incoming mail messages.

Step 5
One last step to do with the new Maildir feature of Qmail is to set up it as the default delivery agent by creating a file named dot-qmail under /etc/qmail directory. The Qmail script initialization file reads this file each time you restart the mail server.

- Create the **dot-qmail** file (touch /etc/qmail/dot-qmail) and add the line:

  ```
  ./Maildir/
  ```

- Change its default mode to (0511/-r-x--x--x) and owned by the super-user 'root':
  ```
  [root@deep /]# chmod 511 /etc/qmail/dot-qmail
  [root@deep /]# chown 0.0 /etc/qmail/dot-qmail
  ```

Step 6
Once the compilation, optimization and installation of the software have been finished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete Qmail and its related source directory, use the following commands:
  ```
  [root@deep /]# cd /var/tmp/
  [root@deep tmp]# rm -rf qmail-version/
  [root@deep tmp]# rm -f qmail-version.tar.gz
  ```

The rm command, as used above, will remove all the source files we have used to compile and install Qmail. It will also remove the Qmail compressed archive from the /var/tmp directory.

Configuring Qmail

After Qmail has been built and installed successfully in your system, your next step is to configure and customize its configuration files to suit your needs.

- ✓ /etc/qmail/control/me: (The Qmail Hostname File)
- ✓ /etc/qmail/control/locals: (The Qmail Local File)
- ✓ /etc/qmail/control/rcpthosts: (The Qmail rcpthost File)
- ✓ /etc/qmail/control/defaultdomain: (The Qmail Defaultdomain File)
- ✓ /etc/qmail/control/plusdomain: (The Qmail Plusdomain File)
- ✓ /etc/init.d/qmail: (The Qmail Initialization File)

/etc/qmail/control/me: The Qmail Hostname Configuration File

All files under /etc/qmail/control directory are configuration files for the Qmail system. Qmail can run with just one control file named "me" which should contains the **F**ully **Q**ualified **D**omain **N**ame (FQDN) of the current host. This file "me" is used as the default for other hostname-related control files. Usually you don't have to change this file "me" since it's already contains your **F**ully **Q**ualified **D**omain **N**ame for Qmail to work, otherwise if it doesn't exist, create it and add your fully qualified domain name (i.e. smtp.domain.com) inside it.

- Edit the **me** file (vi /etc/qmail/control/me) and add:

```
smtp.domain.com
```

/etc/qmail/control/locals: The Qmail Locals Configuration File

The Qmail configuration file locals can be used to handle a list of domain names that the current host receives mail for, one per line. Qmail will know through the content of this file which addresses it should deliver locally.

This file becomes important when you configure Qmail as a Central Mail Hub Server. If you want to configure your Qmail software to run as a Null Client Mail Server, you will need to remove the default value in this file, which is the **F**ully **Q**ualified **D**omain **N**ame (FQDN) of the current host again. See later in this chapter for more information about running Qmail as a Null Client Mail Server.

- Edit the **locals** file (vi /etc/qmail/control/locals) and add:

```
smtp.domain.com
```

/etc/qmail/control/rcpthosts: The Qmail rcpthosts File

This file rcpthosts specifies which domains are allowed to use the Qmail Mail Server. If a domain is not listed in rcpthosts then qmail-smtpd will reject any envelope recipient address. To summarize, Qmail will know through the contents of this file which messages it should accept from remote systems to deliver.

By default with Qmail, relaying is turned off and you must populate the rcpthosts file with the **F**ully **Q**ualified **D**omain **N**ame of all authorized hosts on your network. As for the Exim relaydomains file, one use for such a file might be to declare all **F**ully **Q**ualified **D**omain **N**ame that are local to your network. If your **F**ully **Q**ualified **D**omain **N**ame is "smtp.domain.com", you have to add it into this file for Qmail to work.

You don't need to list all the servers on your network in this file only your **F**ully **Q**ualified **D**omain **N**ame. Again, I repeat, there is no need to list, for example "www.domain.com", or "ftp.domain.com", or "something.domain.com", etc into this file but JUST the **F**ully **Q**ualified **D**omain **N**ame of the server on which Qmail is running "smtp.domain.com".

For virtual hosting, we will also use this file to list all virtual domains hosted on our mail server. See later in this chapter for more information about virtual domain hosting with Qmail.

- Edit the **rcpthosts** file (vi /etc/qmail/control/rcpthosts) and add:

```
smtp.domain.com
```

/etc/qmail/control/defaultdomain: The Qmail defaultdomain File

The defaultdomain file is used by Qmail to add the domain name listed in the file "defaultdomain" to any host name without dots. Usually you don't have to change the default information (i.e. domain.com) listed in this file "defaultdomain".

- Edit the **defaultdomain** file (vi /etc/qmail/control/defaultdomain) and add:

```
domain.com
```

/etc/qmail/control/plusdomain: The Qmail plusdomain File

The plusdomain file is used by Qmail to add the domain name listed in the file "plusdomain" to any host name that ends with a plus sign. Usually you don't have to change the default information (i.e. domain.com) listed in this file "plusdomain".

- Edit the **plusdomain** file (vi /etc/qmail/control/plusdomain) and add:

```
domain.com
```

/etc/init.d/qmail: The Qmail Initialization File

The /etc/init.d/qmail script file is responsible for automatically starting and stopping the Qmail server. Please note that the following script is only suitable for Linux operating systems that use System V. If your Linux system uses some other method, like BSD, you'll have to adjust the script below to make it work for you.

Step 1
Create the **qmail** script file (touch /etc/init.d/qmail) and add the following lines:

```
#!/bin/bash

# This shell script takes care of starting and stopping Qmail.
#
# chkconfig: 2345 80 30
# description:  Qmail is a small, fast, secure Mail Transport Agent, which \
#               is the program that moves mail from one machine to another.
#
# processname: qmail-send

# Source function library.
. /etc/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

# If Qmail is not available stop now.
[ -f /usr/bin/qmail-send ] || exit 0

RETVAL=0
prog="Qmail"

start() {
        echo -n $"Starting $prog: "
```

```
        qmail-start "`cat /etc/qmail/dot-qmail`" splogger qmail &
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/qmail
        return $RETVAL
}

stop() {
        echo -n $"Shutting down $prog: "
        killproc qmail-send
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/qmail
        return $RETVAL
}

# See how we were called.
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  queue)
        qmail-qstat
        qmail-qread
        ;;
  status)
        status qmail-send
        RETVAL=$?
        ;;
  restart)
        stop
        start
        RETVAL=$?
        ;;
  condrestart)
        if [ -f /var/lock/subsys/qmail ]; then
                stop
                start
                RETVAL=$?
        fi
        ;;
  *)
        echo $"Usage: $0 {start|stop|queue|status|restart|condrestart}"
        exit 1
esac
exit $RETVAL
```

Step 2
Once the qmail script file has been created, it is important to make it executable, change its default permissions, create the necessary links and then start it. Making this file executable will allow the system to run it, changing its default permission to allow only the root user to change it, is for security reasons, and the creation of the symbolic links will let the process control initialization of Linux, which is in charge of starting all the normal and authorized processes that need to run at boot time on your system, start the program automatically for you at each system reboot.

- To make this script executable and to change its default permissions, use the commands:
```
[root@deep /]# chmod 700 /etc/init.d/qmail
[root@deep /]# chown 0.0 /etc/init.d/qmail
```

- To create the symbolic rc.d links for Qmail, use the following commands:
```
[root@deep /]# chkconfig --add qmail
[root@deep /]# chkconfig --level 2345 qmail on
```

- To start Qmail software manually, use the following command:
```
[root@deep /]# /etc/init.d/qmail start
Starting Qmail:           [OK]
```

Testing Qmail

Once our mailer software is configured and started, we have to run some recommended tests to make sure Qmail is working correctly on our system. The tests should all complete successfully with no problems or you will eventually lose mail messages.

To be able to successfully run the tests, we have to be the super-user "root" and execute all tests on the terminal of the server.

Test 1 - Verifying that Qmail recognizes a local mailbox:
In this test, we should verify that Qmail can recognize a local mailbox on the system. We send ourselves an empty message. If everything is ok, the message will show up immediately in your mailbox.

- To verify that Qmail can recognize a local mailbox, use the following command:
```
[root@deep /]# echo to: sysadmin | /usr/bin/qmail-inject
```

Test 2 - Verifying that Qmail recognizes a nonexistent local address:
In this test, we should verify that Qmail can recognize a nonexistent local address. We send a message to a nonexistent local address on the system. If everything is ok, you will now have a bounce message in your mailbox.

- To verify that Qmail can recognizes a nonexistent address, use the following command:
```
[root@deep /]# echo to: nonexistent | /usr/bin/qmail-inject
```

Test 3 - Verifying that Qmail can deliver mail to a remote email address:
In this test, we should verify that Qmail can deliver mail to a remote email address on the Internet. To do it, we send an empty message to one of our accounts or friend on another machine on the Internet. If everything is ok, the message will show up immediately in the mailbox on the other system.

- To verify that Qmail can deliver mail to a remote address, use the following command:
 `[root@deep /]# echo to: friend@hotmail.com | /usr/bin/qmail-inject`

If you encounter problems, look at Qmail's log file (/var/log/maillog) to see if there is any relevant information there. Also be sure that your networking setting is correct, that your hostname is working, that your DNS resolves, that your firewall allows SMTP packets to pass, and that your FQDN (**F**ully **Q**ualified **D**omain **N**ame) is available.

Allowing Users to authenticate with Qmail before relaying

An open relay mail server is very dangerous and the preferred method for spammers to abuse your system. If you don't take the time to protect your server with anti relay features, sooner or later spammers will use you, and you will be banned on ORBS and other spam fighting lists.

Qmail is built by default with some Anti-Relay features enabled through its rcpthosts file. This means that any hosts listed in this file will be allowed to relay with Qmail. This is good for a local network, but not for external connections, like roaming users trying to send mail with your server. Therefore, we have to find a solution to the problem. Fortunately, different methods of authentication exist and it is up to us to choose which one we want to use to allow relay.

Some authentication methods, like POP-Before-SMTP, already exist for a few MTA's, but they required you to hack into the source code of your mail software to work and this is not what I really like as a solution. Other methods exist, like using SMTP_AUTH and this is the method that we will use here, since it is compatible with all MUA's on the market.

> **NOTE:** For more information about POP-Before-SMTP, please see: http://whoson.sourceforge.net

With SMTP_AUTH the user must be authenticate when he/she logs onto the server to retrieve or send his/her mail messages. In all cases this is done by connecting to a POP or IMAP server. How the SMTP_AUTH authentication actually works is explained as follows:

1. User connects to his/her POP or IMAP account on the server to retrieve/send mail.
2. User sends mail through their POP/IMAP server, Qmail asks for a username & password.
3. The MUA of the user sends the username & password to Qmail.
4. Qmail compares information for the username & password of the user.
5. If the username & password match, then Qmail allows relaying through the mail server.
6. If the username & password do not match, then Qmail sends an error message.

To achieve this result, we have to install additional software called "checkpassword". This software provides a simple, uniform password-checking interface to all root applications and it is suitable for use by applications such as Anti-Relay or pop3d with Qmail. Without it, we cannot implement Anti-Relay features on the mail server. Also, we have to run qmail-popup and qmail-pop3d, two programs that comes with Qmail, because we have to log in on the POP server to read or send mails.

Installing the `checkpassword` software:

Procedures to allow `SMTP_AUTH` to run with `Qmail` are not difficult to accomplish. Since we should already have patched our `Qmail` source code with `qmail-smtpd-auth`, all we have to do now is to install the "`checkpassword`" program with `Qmail`. `Checkpassword` reads the username and password of the users who want to relay, then looks them up in `/etc/passwd`, verifies them, and then allows the relay if the information is correct. This is possible because we have patched `Qmail` with the `qmail-smtpd-auth` software previously in this chapter.

Step 1
First, we have to get the "`checkpassword`" program from the `qmail` website (http://pobox.com/~djb/checkpwd.html) and copy it to the `/var/tmp` directory of our Linux system and change to this location before expanding the archive. After that, we have to move into the newly created `checkpassword` directory and perform the following steps to compile and optimize it.

```
[root@deep /]# cp checkpassword-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf checkpassword-version.tar.gz
[root@deep tmp]# cd checkpassword-0.90
```

Step 2
Before going into the compilation of the program, we'll edit the `conf-cc` file and change the default compiler flags to fit our own `CPU` architecture for better performance.

- Edit the **`conf-cc`** file (`vi conf-cc`) and change the line:

  ```
  cc -O2
  ```

 To read:

  ```
  gcc -O2 -march=i686 -funroll-loops
  ```

Step 3
Now, we must make a list of files on the system before you install the software and one afterwards then compare them using the **`diff`** utility to find out what files are placed where and finally install `checkpassword` in the system.

```
[root@deep checkpassword-0.90]# make
[root@deep checkpassword-0.90]# cd
[root@deep root]# find /* > CheckPass1
[root@deep root]# cd /var/tmp/checkpassword-0.90/
[root@deep checkpassword-0.90]# make setup check
[root@deep checkpassword-0.90]# cd
[root@deep root]# find /* > CheckPass2
[root@deep root]# diff CheckPass1 CheckPass2 > CheckPass-Installed
```

Adding the required SMTP_AUTH parameters to the qmail initialization file:

For Qmail to support SMTP_AUTH, we have to change the default /etc/init.d/qmail initialization file. This is required because Qmail uses tcpserver to start and we have to define some new parameters with tcpserver to make Qmail run with SMTP_AUTH.

What we do inside this startup file is simple, we add a new line related to qmail-popup and qmail-pop3d program to start Qmail with the POP service, next, we change the way qmail-smtpd is run by adding new parameters for the checkpassword utility to perform authentication when SMTP is invoked to send mail. Text in bold is what I've added/changed from the default /etc/init.d/qmail script file.

Step 1
Edit the **qmail** script file (vi /etc/init.d/qmail) and add/change the following lines:

```
#!/bin/bash

# This shell script takes care of starting and stopping Qmail.
#
# chkconfig: 2345 80 30
# description:  Qmail is a small, fast, secure Mail Transport Agent, which \
#               is the program that moves mail from one machine to another.
#
# processname: qmail-send

# Source function library.
. /etc/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

# If Qmail is not available stop now.
[ -f /usr/bin/qmail-send ] || exit 0

RETVAL=0
prog="Qmail"

start() {
        echo -n $"Starting $prog: "
        qmail-start "`cat /etc/qmail/dot-qmail`" splogger qmail &

        # Here we start qmail-smtpd with AUTH support.
        tcpserver -p -c 1024 -DRHl localhost 0.0.0.0 25 /usr/bin/tcp-env \
        tcp-env /usr/bin/qmail-smtpd /bin/checkpassword /bin/true &

        # Here we start qmail-pop3d with AUTH support.
        tcpserver -c 1024 -DRHl localhost 0.0.0.0 110 /usr/bin/qmail-popup \
        `hostname -f` /bin/checkpassword /usr/bin/qmail-pop3d Maildir &
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/qmail
        return $RETVAL
}

stop() {
        echo -n $"Shutting down $prog: "
        killproc qmail-send
        killproc tcpserver
```

```
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/qmail
        return $RETVAL
}

# See how we were called.
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  queue)
        qmail-qstat
        qmail-qread
        ;;
  status)
        status qmail-send
        RETVAL=$?
        ;;
  restart)
        stop
        start
        RETVAL=$?
        ;;
  condrestart)
        if [ -f /var/lock/subsys/qmail ]; then
                stop
                start
                RETVAL=$?
        fi
        ;;
  *)
        echo $"Usage: $0 {start|stop|queue|status|restart|condrestart}"
        exit 1
esac
exit $RETVAL
```

Step 2
Finally, we have to restart `Qmail` daemon for the changes to take effect.

- To restart `Qmail`, use the following command:

```
[root@deep /]# /etc/init.d/qmail restart
Shutting down Qmail:              [OK]
Starting Qmail:                   [OK]
```

WARNING: Your `MUA` must be configured to support `SMTP_AUTH`. In my experience, Netscape works out of the box with `SMTP_AUTH` and you don't need to configure it for this purpose. In the other hand, MS Outlook needs some configuration on your part, you should be sure that the option under your Outlook account named "**`My server requires authentication`**" is checked. See your `MUA` manual for more information about how to enable `SMTP_AUTH`.

Running Qmail with SSL support

This section applies only if you want to run Qmail through an SSL connection. To begin our implementation of SSL into Qmail we will first install a new program called "stunnel" which allows us to provide SSL encryption with Qmail. Next, we will create the necessary certificate keys and add the required SSL parameters to the qmail initialization file. Running Qmail with SSL support is not for everyone. Before we embark on this, we need to first decide whether it is beneficial for us to do so. Some pros and cons are, but most certainly not limited to, the following.

Pros:

- ✓ Client and server of a SMTP connection can be identified.
- ✓ The transmission of e-mail between a client and server utilizing SSL cannot be read and retranslated into plaintext provided a sufficiently secure cipher suite has been negotiated.
- ✓ The plaintext of e-mail between a client and server utilizing SSL cannot be modified by someone, provided a sufficiently secure cipher suite has been negotiated.

Cons:

- ✓ It does not provide end-to-end encryption, since a user can doesn't usually control the whole transmission. This is in contrast to the use of SSL for HTTP: here the user's client (a WWW browser) connects directly to the server that provides the data. E-mail can be transferred via multiple hops of which the sender can control at most the first.
- ✓ It does not provide message authentication, unless the e-mail has been sent directly from the client's (SSL-capable) MUA to the recipients MTA that must record the client's certificate. Even then the message might be faked during local delivery.

To be able to run Qmail with SSL support, we have to install additional software called "stunnel". This software allows us to encrypt arbitrary TCP connections inside SSL and secure non-SSL aware daemons and protocols by having stunnel provide the encryption, requiring no changes to the daemon's code. Without it, we cannot implement the encryption feature on the mail server.

Installing the stunnel software:

There are some SSL patches available on the Internet to run Qmail with SSL support but I don't recommend you to go with this solution, because the SSL patch breaks and conflicts with the Anti-Relay patch we have used previously to run Qmail with Anti-Relay support. The best solution is to go with the "stunnel" program. Here I explain how to get and install stunnel to run with Qmail to provide SSL support for the Qmail SMTP daemon.

Step 1
First, we have to get the "stunnel" program from the stunnel website (http://www.stunnel.org/) and copy it to the /var/tmp directory of our Linux system and change to this location before expanding the archive. After that, we have to move into the newly created stunnel directory and perform the following steps to compile, optimize and install it.

```
[root@deep /]# cp stunnel-3.22.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf stunnel-3.22.tar.gz
[root@deep tmp]# cd stunnel-3.22
```

Step 2
Now we have to configure, compile and optimize it for our system.

- To configure, compile and optimize `stunnel` use the following compilation lines:

```
CFLAGS="-O2 -march=i686 -funroll-loops"; export CFLAGS
./configure \
--prefix=/usr \
--mandir=/usr/share/man \
--with-ssl=/usr \
--with-pem-dir=/usr/share/ssl/certs \
--with-cert-file=/usr/share/ssl/private \
--with-cert-dir=/usr/share/ssl/trusted
```

Step 3
Now, we must make a list of files on the system before you install the software, and one afterwards then compare them using the **diff** utility to find out what files are placed where and finally install `stunnel` in the system.

```
[root@deep stunnel-3.22]# make piddir=/var/run/
[root@deep stunnel-3.22]# cd
[root@deep root]# find /* > Stunnel1
[root@deep root]# cd /var/tmp/stunnel-3.22/
[root@deep stunnel-3.22]# make install
[root@deep stunnel-3.22]# strip /usr/sbin/stunnel
[root@deep stunnel-3.22]# rm -f /usr/share/ssl/certs/stunnel.pem
[root@deep stunnel-3.22]# rm -rf /usr/var
[root@deep stunnel-3.22]# /sbin/ldconfig
[root@deep stunnel-3.22]# cd
[root@deep root]# find /* > Stunnel2
[root@deep root]# diff Stunnel1 Stunnel2 > Stunnel-Installed
```

Creating the necessary `Qmail` certificate keys:

Here I show you how to set up a self-signed certificate to use with `Qmail`, again the principle is the same as for creating a certificate for a Web Server (refer to `OpenSSL` chapter if you have problem creating the certificates).

Step 1
First you have to know the **F**ully **Q**ualified **D**omain **N**ame (`FQDN`) of the Central Mail Hub Server for which you want to request a certificate. When your incoming mail server address is `smtp.domain.com` then the `FQDN` of your Central Mail Hub Server is `smtp.domain.com`.

Step 2
Create a self-signed certificate (`x509` structure) without a pass-phrase. The `req` command creates a self-signed certificate when the `-x509` switch is used. For certificates signed by commercial **C**ertifying **A**uthority (`CA`) like Thawte refer to the `OpenSSL` chapter for the required procedures to follow.

- To create a self-signed certificate, use the following command:

```
[root@deep ssl]# cd /usr/share/ssl
[root@deep ssl]# openssl req -new -x509 -nodes -days 365 -out tmp.pem
Using configuration from /usr/share/ssl/openssl.cnf
Generating a 1024 bit RSA private key
.............................................++++++
...........++++++
writing new private key to 'privkey.pem'
-----
```

```
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [CA]:
State or Province Name (full name) [Quebec]:
Locality Name (eg, city) [Montreal]:
Organization Name (eg, company) [Open Network Architecture]:
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) [smtp.domain.com]:
Email Address [noc@domain.com]:
```

WARNING: Pay special attention to the '`-nodes`' option we have used, in the above command, to create the self-signed certificate. The option '`-nodes`' creates a certificate without a protected pass-phrase, it is very important to create a certificate without a pass-phrase because `Qmail` cannot ask you to enter a password before starting its daemon. Also, be sure that you've entered the `FQDN` (**F**ully **Q**ualified **D**omain **N**ame) of the Central Mail Hub Server when `OpenSSL` prompts you for the "**`Common Name`**".

Step 3
Once the self-signed certificate has been created, we must be sure that the future **`smtp.pem`** file will has both a `RSA PRIVATE KEY` and a `CERTIFICATE` section.

- To include the `CERTIFICATE` section to `RSA PRIVATE KEY`, use the command:
 `[root@deep ssl]#` **`cat tmp.pem >> privkey.pem`**

The above command will include the `CERTIFICATE` file named "`tmp.pem`" to the `RSA PRIVATE KEY` named "`privkey.pem`".

Step 4
Next, we must place the certificate file to its appropriate directory and rename it "`smtp.pem`" for `Qmail` server to use it.

- To place the file into its appropriate directory, use the following command:

```
[root@deep ssl]# mv privkey.pem certs/smtp.pem
[root@deep ssl]# chmod 400 certs/smtp.pem
[root@deep ssl]# chown 0.0 certs/smtp.pem
[root@deep ssl]# rm -f tmp.pem
```

First we move the `privkey` file, which contains both the `RSA PRIVATE KEY` and `CERTIFICATE` sections to the `certs` directory and rename it `smtp.pem` for `Qmail` to use it for `SMTP` protocol. After that, we remove the `tmp.pem` file from our system since it is no longer needed.

Adding the required SSL parameters to the qmail initialization file:

Once the Qmail certificate has been created and moved to the appropriate location, we have to change the default /etc/init.d/qmail initialization file for Qmail to be configured to run with SSL support on the server. This is required because Qmail should use stunnel to start and we have to define some new parameters with stunnel to make Qmail run with SSL.

Below I show you the required lines to add or change in your default qmail initialization file for Qmail to run with SSL support. Text in bold is what we have added or changed to the default Qmail initialization file.

Step 1
Edit the **qmail** script file (vi /etc/init.d/qmail) and add/change the following lines:

```
#!/bin/bash

# This shell script takes care of starting and stopping Qmail.
#
# chkconfig: 2345 80 30
# description:  Qmail is a small, fast, secure Mail Transport Agent, which \
#               is the program that moves mail from one machine to another.
#
# processname: qmail-send

# Source function library.
. /etc/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

# If Qmail is not available stop now.
[ -f /usr/bin/qmail-send ] || exit 0

RETVAL=0
prog="Qmail"

start() {
        echo -n $"Starting $prog: "
        qmail-start "`cat /etc/qmail/dot-qmail`" splogger qmail &

        # Here we start qmail-smtpd with AUTH & SSL support.
        stunnel -n smtp -d 25 -o /var/log/maillog -p \
        /usr/share/ssl/certs/smtp.pem -l /usr/bin/tcp-env -- tcp-env \
        /usr/bin/qmail-smtpd /bin/checkpassword /bin/true

        # Here we start qmail-pop3d with AUTH support.
        tcpserver -c 1024 -DRHl localhost 0.0.0.0 110 /usr/bin/qmail-popup \
        `hostname -f` /bin/checkpassword /usr/bin/qmail-pop3d Maildir &
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/qmail
        return $RETVAL
}

stop() {
        echo -n $"Shutting down $prog: "
        killproc qmail-send
        killproc tcpserver
```

```
        killproc stunnel
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/qmail
        return $RETVAL
}

# See how we were called.
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  queue)
        qmail-qstat
        qmail-qread
        ;;
  status)
        status qmail-send
        RETVAL=$?
        ;;
  restart)
        stop
        start
        RETVAL=$?
        ;;
  condrestart)
        if [ -f /var/lock/subsys/qmail ]; then
                stop
                start
                RETVAL=$?
        fi
        ;;
  *)
        echo $"Usage: $0 {start|stop|queue|status|restart|condrestart}"
        exit 1
esac
exit $RETVAL
```

Step 2
Finally, we have to restart `Qmail` daemon for the changes to take effect.

- To restart `Qmail`, use the following command:

```
[root@deep /]# /etc/init.d/qmail restart
Shutting down Qmail:              [OK]
Starting Qmail:                   [OK]
```

Running Qmail with Virtual Hosts support

This section applies only if you want to run Qmail with virtual hosts support. Virtual hosts, or if you prefer virtual mail, is when you provide hosting on the same server for many domains and want to use Qmail as the mail server for all the domains hosted on the server. This is very interesting for hosting companies, since they don't need to dedicate servers for virtual mail or even run more than one mail server to provide a mail service for all their virtual domains.

Creating the necessary virtualdomains file:

The /etc/qmail/control/virtualdomains file is used to define virtual aliasing mail accounts to virtual domains hosted on your server. You should use it every time you need to set up a virtual domain and define aliases for virtual mail accounts on your system.

Step 1
By default, the virtualdomains file does not exist, so we have to create it.

- Create the virtualdomains file (touch /etc/qmail/control/virtualdomains) and add all virtual domains and aliases for virtual mail accounts inside it:

```
# This file must contains any email aliases for your virtual hosts users.
# For example, you do:
#
# virtual1.com:mark
# virtual2.com:john
#
# N.B.: Qmail must be restarted after this file is modified.
#
virtual1.com:mark
virtual2.com:john
```

In the above example, all email messages addressed to whatever@virtual1.com will be delivered locally to mark@virtual1.com on this virtual domain and any email messages addressed to whatever@virtual2.com will be delivered locally to john@virtual2.com on this virtual domain. The both domains reside on the same server.

To complete the above setup, Mark should create in his home directory "/home/mark" a new file called ".qmail-default" to catch all the possible addresses, or create another file called ".qmail-info" to catch info@virtual1.com, and so on. This is true for the user John and any other users on the system.

Step2
Now, set the permissions mode of the virtualdomains file to be (0644/-rw-r--r--) and owned by the super-user 'root' for security reasons.

- To change the permission mode and ownership of the virtualdomains file, use:

```
[root@deep /]# chmod 644 /etc/qmail/control/virtualdomains
[root@deep /]# chown root.root /etc/qmail/control/virtualdomains
```

NOTE: For virtual domains to work, you have to be sure that the MX record has been set in the primary and secondary DNS servers for the virtual domain. If an MX record doesn't exist, then set it up in your DNS servers before configuring virtual hosts. This is true for all mail server software.

Allowing `Virtual Hosts` to relay:

Remember, that `Qmail` by default does not allow you to relay without the proper authentication, this is true for virtual domains too. Your have to be sure the virtual domain in question is added into the `rcpthosts` file of `Qmail` to be allowed to relay. This is very important or relaying will be denied.

Step 1
Add all the virtual domains you want `Qmail` to relay into the `rcpthosts` file for relaying to work.

- Edit your **`rcpthosts`** file (`vi /etc/qmail/control/rcpthosts`) and add:

```
# This file handle all domains from which relaying is allowed.
# By default we include the locahost of the server or nothing will work.
# Virtual Domains must be added to this list or relaying will be denied.
# N.B.: Qmail must be restarted after this file is modified.
#
smtp.domain.com
virtual1.com
virtual2.com
```

WARNING: If a virtual domain is listed in the `/etc/qmail/control/locals` file, then virtualdomains does NOT apply and will NOT work. This means that you ONLY need to add the virtual domain name in question into the `virtualdomains` and `rcpthosts` files.

Step 2
Now, we have to restart the `Qmail` daemon for the changes to take effect.

- To restart `Qmail`, use the following command:

```
[root@deep /]# /etc/init.d/qmail restart
Shutting down Qmail:              [OK]
Starting Qmail:                   [OK]
```

Installing the `fastforward` software:

With the above configuration, virtual hosts will work fine but if you want for example to forward one of you local virtual user accounts to an external email account like "`billy@hotmail.com`", you will have some more work to do. To solve the problem, we can install `Qmail` software called "`fastforward`". `Fastforward` handles `Qmail` forwarding according to a `cdb` database. It can create forwarding databases from sendmail-style `/etc/aliases` or from user-oriented virtual-domain tables.

Step 1
First, we have to get the "`fastforward`" program (http://cr.yp.to/fastforward.html) and copy it to the `/var/tmp` directory of our system and then change to this location before expanding the archive. After that, we have to move into the newly created `fastforward` directory and perform the following steps to compile and optimize it.

```
[root@deep /]# cp fastforward-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf fastforward-version.tar.gz
[root@deep tmp]# cd fastforward-0.51
```

Step 2
Before going into the compilation of the program, we'll edit the `conf-cc` file and change the default compiler flags to fit our own `CPU` architecture for better performance.

- Edit the **`conf-cc`** file (`vi conf-cc`) and change the line:

  ```
  cc -O2
  ```

 To read:

  ```
  gcc -O2 -march=i686 -funroll-loops
  ```

Step 3
Now, we must make a list of files on the system before you install the software, and one afterwards then compare them using the **`diff`** utility to find out what files are placed where and finally install `fastforward` in the system.

```
[root@deep fastforward-0.51]# make
[root@deep fastforward-0.51]# cd
[root@deep root]# find /* > Fast1
[root@deep root]# cd /var/tmp/fastforward-0.51/
[root@deep fastforward-0.51]# make setup check
[root@deep fastforward-0.51]# rm -rf /var/qmail/doc/
[root@deep fastforward-0.51]# cd
[root@deep root]# find /* > Fast2
[root@deep root]# diff Fast1 Fast2 > Fast-Installed
```

Step 4
Once the `fastforward` software is installed, we have to create the `aliases` file and populate it with some aliases to make the program to work with `Qmail`.

- Create the **`aliases`** file (`touch /etc/aliases`) and add the following lines. Below is what we recommend you set.

```
# The following aliases are required by the mail RFCs 2821 and 2822.
# At least, you should set "postmaster" to the address of a HUMAN
# who deals with this system's mail problems.
#
postmaster:     sysadmin@localhost
mailer-daemon:  postmaster
root:           postmaster

# It is a good idea to redirect any messages sent to system accounts
# so that they don't just get ignored.
#
bin:            root
daemon:         root
decode:         root
dumper:         root
sync:           root
mail:           root
pop:            root
uucp:           root
ftp:            root
nobody:         root
www:            root
manager:        root
```

```
named:          root
postgres:       root
mysql:          root
squid:          root
amavis:         root
operator:       root
abuse:          root
hostmaster:     root
webmaster:      root
```

NOTE: Please, don't forget to change "postmaster" to the email address of your real system administrator on your network. The above "sysadmin@localhost" is an example, therefore change it. Your aliases file will be probably far more complex, but even so, note how the example shows the minimum form of aliases.

Step 5
Now, set the permission mode of the aliases file to be (0644/-rw-r--r--) and owned by the super-user 'root' for security reasons.

- To change the permission mode and ownership of "aliases" file, use the commands:
```
[root@deep /]# chmod 644 /etc/aliases
[root@deep /]# chown 0.0 /etc/aliases
```

Step 6
Finally, we have to create a new file called ".qmail-default" under the /etc/qmail/alias directory. This file is used by Qmail to automatically read the aliases file each time it need it.

- Create the **.qmail-default** file (touch /etc/qmail/alias/.qmail-default) and add the following line.

```
| fastforward -d /etc/aliases.cdb
```

Step 7
Now, set the permission mode of the .qmail-default file to be (0644/-rw-r--r--) and owned by the super-user 'root' with group permission set to "nofiles" user for security reason.

- To change the permission mode and ownership of ".qmail-default" file, use:
```
[root@deep /]# chmod 644 /etc/qmail/alias/.qmail-default
[root@deep /]# chown root.nofiles /etc/qmail/alias/.qmail-default
```

At this stage of your work, the program is working and you can forward mail according to what you define inside the /etc/aliases file for your local or virtual users.

NOTE: Please note that a program called "newaliases" is available with the program to update the aliases database as we do with Exim. Don't forget to use it each time you change information inside the /etc/aliases file.

Running `Qmail` as a Null Client Mail Server

Now we have explained how to configure `Qmail` to send/receive mail to both the internal and external networks. This kind of configuration is also known as a Central Mail Hub Server and it's useful when you need to have a fully operational mail server. In some cases we simply don't need such a complete setup because the server on which we want to run `Qmail` is not configured to be principally a mail server, but to run other kinds of services like web, ftp, dns, gateway, etc. On these kinds of servers, we don't need to run a full operational mail server capable of receiving and sending mail everyway, because the only mail generated on these servers should be mail dedicated to system account on the system.

For these servers, we can configure `Qmail` to run as a Null Client Mail Server. A Null Client Mail Server is a mail server that doesn't listen for incoming connections to the server for receiving mail. It can only send all locally generated mails dedicated to local system accounts to where you want them to go. This means that a Null Client Mail Server can only deliver locally generated mails to a Central Mail Hub Server of your choice.

This is highly useful to improve the security of your mail server, because it does not listen for incoming mail connections and spammers cannot use it to forward mail. Here is an overview of both possible configurations for a mail server:

A Central Mail Hub Server configuration:
Instead of having each individual server or workstation in a network handle its own mail, it can be advantageous to have powerful central server that handles all mail. Such a server is called a Central Mail Hub Server. The advantage of a Central Mail Hub Server is:

- ✓ Can receive or send mail to everywhere.
- ✓ Allow mail users to connect to their mail account to send or get mail.
- ✓ Capable to receive and manage mail coming from null client mail servers.

A Null Client Mail Server configuration:
Mail service is indispensable for all types of server, even if the task of the server in question is not to process mail, because at least we should be able to get important messages generated locally and addressed to the local system account (`posmaster`) on the server.

A null client mail server never receives mail directly from the outside world and relay (send) all their mail through the Central Mail Hub Server. In this way, important messages addressed to the local system account can be delivered remotely to the Central Mail Hub Server for verification. The advantage of a Null Client Mail Server is:

- ✓ No mail is sent directly to a local user account on the system.
- ✓ All mails are sent (forwarded) to the Central Mail Hub Server.
- ✓ No client's name needs to be known to the outside world.
- ✓ `Qmail` daemon do not need to listen for incoming mail connection.

Creating the necessary `smtproutes` file:

The `/etc/qmail/control/smtproutes` file is used to define the **F**ully **Q**ualified **D**omain **N**ame (`FQDN`) of the remote Central Mail Hub Server which should handle all mail for our Null Client Mail Server. You should create it every time you need to set up Null Client Mail Server on your system.

Step 1
By default, the `smtproutes` file does not exist, so we have to create it.

- Create the **`smtproutes`** file (`touch /etc/qmail/control/smtproutes`) and add inside it, the **F**ully **Q**ualified **D**omain **N**ame (`FQDN`) of the remote Central Mail Hub Server which should handle all mail for our Null Client Mail Server:

  ```
  :mail.domain.com
  ```

In the above example, <`:mail.domain.com`> is the domain name of our Central Mail Hub Server where we want to send all outgoing mail messages. The "`:`" means transfer all outgoing mail through the "`domain.com`" domain name.

Step2
Now, set the permission mode of the `smtproutes` file to be (`0644/-rw-r--r--`) and owned by the super-user '`root`' for security reasons.

- To change the permission mode and ownership of the `smtproutes` file, use:
  ```
  [root@deep /]# chmod 644 /etc/qmail/control/smtproutes
  [root@deep /]# chown root.root /etc/qmail/control/smtproutes
  ```

Step 3
Finally, we must disable local delivery. This is important because we want to forward all local mail to the Central Mail Hub Server. To archive this result, we have to remove the `locals` file of `Qmail` on the Null Client Mail Server.

- To remove the **`locals`** file of `Qmail`, use the following command:
  ```
  [root@deep /]# rm -f /etc/qmail/control/locals
  ```

WARNING: It's important to be sure that the `MX` record is set up properly in your `DNS` (**D**omain **N**ame **S**erver) server before you do this. Also be sure that `ucspi-tcp`, `fastforward` and `checkpassword` packages are not installed. A `Qmail` Null Client Mail Server doesn't need this software to be installed on your server.

Updating all of your `.qmail-*` alias files:

Once the `smtproutes` file has been set, we have to edit all of our `.qmail-*` files and change the user listed inside the files to make it point to a remote user who should receive system account messages sent by the Null Client Mail Server to the Central Mail Hub Sever.

- This can be done with the following commands.

```
[root@deep qmail-1.03]# cd ~alias
[root@deep alias]# echo noc@domain.com > .qmail-postmaster
[root@deep alias]# echo noc@domain.com > .qmail-mailer-daemon
[root@deep alias]# echo noc@domain.com > .qmail-root
```

Here, we instruct `Qmail` to send all messages intended for "`postmaster`", "`mailer-daemon`" or the super-user "`root`" to a remote non-privileged user account named `noc` at `domain.com`.

Making `Qmail` to not listen for incoming mail connection:

When `Qmail` is configured to run as a Null Client Mail Server, it should not listen for incoming connections on your server. This is very important, for security reasons, and we must edit our default `/etc/init.d/qmail` file and make the change.

For `Qmail` to run as a fully operational Central Mail Hub Server on the system, it should be configured to start `qmail-smtpd`, `qmail-popup` and `qmail-pop3d`. This is what we do by default when we install the software. We must change this when `Qmail` is running as a Null Client Mail Server.

Below I show you the lines you need to add or change to your default `qmail` initialization file for running `Qmail` as a Null Client Mail Server. Text in bold is what we have added/changed to the default `Qmail` initialization file.

Step 1
Edit the **`qmail`** script file (`vi /etc/init.d/qmail`) and add/change the following lines:

```
#!/bin/bash

# This shell script takes care of starting and stopping Qmail.
#
# chkconfig: 2345 80 30
# description:  Qmail is a small, fast, secure Mail Transport Agent, which \
#               is the program that moves mail from one machine to another.
#
# processname: qmail-send

# Source function library.
. /etc/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

# If Qmail is not available stop now.
[ -f /usr/bin/qmail-send ] || exit 0

RETVAL=0
prog="Qmail"

start() {
```

```
        echo -n $"Starting $prog: "
        qmail-start "`cat /etc/qmail/dot-qmail`" splogger qmail &
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/qmail
        return $RETVAL
}

stop() {
        echo -n $"Shutting down $prog: "
        killproc qmail-send
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/qmail
        return $RETVAL
}

# See how we were called.
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  queue)
        qmail-qstat
        qmail-qread
        ;;
  status)
        status qmail-send
        RETVAL=$?
        ;;
  restart)
        stop
        start
        RETVAL=$?
        ;;
  condrestart)
        if [ -f /var/lock/subsys/qmail ]; then
                stop
                start
                RETVAL=$?
        fi
        ;;
  *)
        echo $"Usage: $0 {start|stop|queue|status|restart|condrestart}"
        exit 1
esac
exit $RETVAL
```

Step 2
Finally, we have to restart `Qmail` daemon for the changes to take effect.

- To restart `Qmail`, use the following command:

```
[root@deep /]# /etc/init.d/qmail restart
Shutting down Qmail:              [OK]
Starting Qmail:                   [OK]
```

Running Qmail as a Mini-Qmail Mail Server

This section applies only if you chose to install and use Qmail as a Mini-Qmail Mail Server on your system. We have already successfully configured Qmail to run as a Null Client Mail Server previously. Here we'll show you how to run it in a much more secure manner again.

The difference with the previous Null Client Mail Server configuration of Qmail is the fact that in this set up, the Central Mail Hub Server at the other end must absolutely be a Qmail server. This kind of configuration is known as a Mini-Qmail installation and provides the following advantages.

1. A Mini-Qmail installation doesn't do any local delivery.
2. A Mini-Qmail installation runs with the same privileges as the user sending mail.
3. A Mini-Qmail installation doesn't need entries in /etc/group or /etc/passwd files.
4. A Mini-Qmail installation doesn't have a queue.
5. A Mini-Qmail installation doesn't receive incoming mail.

To achieve this, we have to configure the Mini-Qmail Server and the Central Mail Hub Server responsible to receive mail from the Mini-Qmail Server to run with the Qmail QMQP (**Q**uick **M**ail **Q**ueuing **P**rotocol) protocol. This is extremely important because QMQP is the protocol used by a Mini-Qmail to run on your server and the QMQP protocol is also what should be used on the remote Central Mail Hub Server to recognize and accept mail messages coming from the Mini-Qmail Server.

This means that we have to run the QMQP protocol on the Mini-Qmail Server and the Central Mail Hub Server. Also we must be sure that our firewall allows the QMQP traffics to pass on both parts. On the Mini-Qmail Server you have to open port number 209 and on the Central Mail Hub Server, you have to open port number 628.

Removing all unnecessary directories and files:

With a Mini-Qmail mail server setup, there are many directories and files to remove from your default installation because they are no longer required.

Step 1
With Qmail running as a Mini-Qmail mail server, you don't need /etc/qmail/alias. A Mini-Qmail installation doesn't deliver mail locally, therefore we can safety remove the entire alias directory from our system and the fastforward software if it is installed.

- This can be done with the following commands:

```
[root@deep /]# rm -rf /var/qmail/alias
[root@deep /]# rm -rf /etc/qmail/alias
```

Step 2
A Mini-Qmail configuration doesn't need any Qmail entries in /etc/group and /etc/passwd. It runs with the same privileges as the user sending mail and doesn't have any of its own files, therefore we can safety remove any Qmail users and group names we have created in this chapter in the passwd and group file and the checkpassword software if it is installed.

- This can be done with the following commands:

```
[root@deep /]# userdel alias
[root@deep /]# userdel qmaild
[root@deep /]# userdel qmaill
[root@deep /]# userdel qmailp
[root@deep /]# userdel qmailq
[root@deep /]# userdel qmailr
[root@deep /]# userdel qmails
```

```
[root@deep /]# groupdel qmail
[root@deep /]# groupdel nofiles
```

Step 3
A Mini-Qmail configuration doesn't need a long-running queue manager. This means that we can safety remove the entire `/var/qmail/queue` directory from our system.

- This can be done with the following command:
```
[root@deep /]# rm -rf /var/qmail/queue
```

Step 4
A Mini-Qmail configuration doesn't receive incoming mail. This means that we can safety remove the entire `/etc/init.d/qmail` initialization script file from our system and the `ucspi-tcp` software if it is installed.

- This can be done with the following commands:
```
[root@deep /]# chkconfig --level 2345 qmail off
[root@deep /]# chkconfig --del qmail
[root@deep /]# rm -f /etc/init.d/qmail
```

Step 5
Since we run a highly secure and fast Mini-Qmail setup, there are many `Qmail` binaries that should be removed from the `/var/qmail/bin` directory of the system since they're no longer required in this configuration.

- Remove all non needed `qmail` binaries from the system with the following commands:
```
[root@deep /]# rm -f /var/qmail/bin/bouncesaying
[root@deep /]# rm -f /var/qmail/bin/condredirect
[root@deep /]# rm -f /var/qmail/bin/except
[root@deep /]# rm -f /var/qmail/bin/preline
[root@deep /]# rm -f /var/qmail/bin/qbiff
[root@deep /]# rm -f /var/qmail/bin/qmail-clean
[root@deep /]# rm -f /var/qmail/bin/qmail-getpw
[root@deep /]# rm -f /var/qmail/bin/qmail-local
[root@deep /]# rm -f /var/qmail/bin/qmail-lspawn
[root@deep /]# rm -f /var/qmail/bin/qmail-newmrh
[root@deep /]# rm -f /var/qmail/bin/qmail-newu
[root@deep /]# rm -f /var/qmail/bin/qmail-pw2u
[root@deep /]# rm -f /var/qmail/bin/qmail-qmqpd
[root@deep /]# rm -f /var/qmail/bin/qmail-qread
[root@deep /]# rm -f /var/qmail/bin/qmail-qstat
[root@deep /]# rm -f /var/qmail/bin/qmail-queue
[root@deep /]# rm -f /var/qmail/bin/qmail-remote
[root@deep /]# rm -f /var/qmail/bin/qmail-rspawn
[root@deep /]# rm -f /var/qmail/bin/qmail-qmtpd
[root@deep /]# rm -f /var/qmail/bin/qmail-send
[root@deep /]# rm -f /var/qmail/bin/qmail-smtpd
[root@deep /]# rm -f /var/qmail/bin/qmail-start
[root@deep /]# rm -f /var/qmail/bin/qmail-tcpok
[root@deep /]# rm -f /var/qmail/bin/qmail-tcpto
[root@deep /]# rm -f /var/qmail/bin/qreceipt
[root@deep /]# rm -f /var/qmail/bin/qsmhook
[root@deep /]# rm -f /var/qmail/bin/splogger
[root@deep /]# rm -f /var/qmail/bin/tcp-env
```

Step 6
One last step is to create a symbolic link to qmail-qmqpc from /usr/bin/qmail-queue. The qmail-qmqpc offers the same interface as qmail-queue, but it gives the message to a QMQP server instead of storing it locally.

- To create the symbolic link, use the following commands:
```
[root@deep /]# cd /usr/bin
[root@deep bin]# ln -s qmail-qmqpc /usr/bin/qmail-queue
```

Creating the necessary qmqpservers file:

Remember that a Mini-Qmail configuration setup transfers all local mail messages to the remote Central Mail Hub Server via the QMQP protocol and not the SMTP protocol. This means that the remote Central Mail Hub Server should runs a copy of the Qmail's QMQP protocol to be able to receive mail from the Mini-Qmail server.

On the Mini-Qmail server, we have to inform the system about the address of the remote Central Mail Hub Server responsible for accepting mail coming from the Mini-Qmail server. This is possible by adding the IP address (NOT the FQDN) of the Central Mail Hub Server that runs a copy of one QMQP daemon into the /etc/qmail/control/qmqpservers file.

Step 1
By default, the qmqpservers file does not exist, we have to create it.

- Create the **qmqpservers** file (touch /etc/qmail/control/qmqpservers) and add the IP address of the remote Central Mail Hub Server which should receive all mail for our Mini-Qmail Server:

```
1.2.3.4
```

In the above example, <1.2.3.4> is the IP address of our Central Mail Hub Server where we want to send all outgoing mail messages. It is important to use IP address here and not domain name or FQDN. This is very important or it will not work.

Step2
Now, set the permission mode of the qmqpservers file to be (0644/-rw-r--r--) and owned by the super-user 'root' for security reasons.

- To change the permission mode and ownership of the qmqpservers file, use:
```
[root@deep /]# chmod 644 /etc/qmail/control/qmqpservers
[root@deep /]# chown root.root /etc/qmail/control/qmqpservers
```

Creating the necessary idhost file:

The /etc/qmail/control/idhost file is used to define the **F**ully **Q**ualified **D**omain **N**ame of the server which runs the Mini-Qmail server. This file is used by Qmail to generate Message-ID's to avoid any risk of collision.

Step 1
By default, the idhost file does not exist and we have to create it.

- Create the **idhost** file (touch /etc/qmail/control/idhost) and add the **F**ully **Q**ualified **D**omain **N**ame of the server which runs the Mini-Qmail server inside it:

 smtp.domain.com

In the above example, the "smtp.domain.com" is the **F**ully **Q**ualified **D**omain **N**ame of the server, which runs the Mini-Qmail server.

Step2
Now, set the permission mode of the idhost file to be (0644/-rw-r--r--) and owned by the super-user 'root' for security reasons.

- To change the permission mode and ownership of the idhost file, use:

```
[root@deep /]# chmod 644 /etc/qmail/control/idhost
[root@deep /]# chown root.root /etc/qmail/control/idhost
```

Step 3
At this stage of your configuration, the Mini-Qmail server is configured and ready to run on your system. One last thing to check is to be sure that the /etc/qmail/control/locals, /etc/qmail/control/smtproutes and /etc/qmail/control/rcpthosts files do not exist on your server; if these files are present, remove them. This is important.

- This can be done with the following commands:

```
[root@deep /]# rm -f /etc/qmail/control/locals
[root@deep /]# rm -f /etc/qmail/control/rcpthosts
[root@deep /]# rm -f /etc/qmail/control/smtproutes
```

Running the QMQP daemon on the Central Mail Hub Server:

Now, our Mini-Qmail Server is capable of sending all locally generated mail messages to the Central Mail Hub Server for delivery, but for the remote Central Mail Hub Server to be able to recognize and accept incoming mail from the Mini-Qmail Server, we have to start the QMQP daemon to inform Qmail to listen for incoming QMQP connections on the server.

Step 1
This is possible by adding the following line into your default qmail initialization script file or into your rc file on the Central Mail Hub Server.

- Use the following line to start the QMQP daemon:

```
tcpserver -p -c 128 -DRHl localhost 0.0.0.0 628 /usr/bin/qmail-qmqpd &
```

NOTE: Don't forget to allow traffic through port 628 into your firewall script file for the qmail-qmqpd daemon to work properly.

Running qmail-pop3d with SSL support

This section applies only if you want to run qmail-pop3d through an SSL connection. To be able to run qmail-pop3d with SSL support, we have to be sure the software "stunnel" is already installed on our system. If this is not the case, then refer to the previous section in this chapter where we discuss it. Stunnel allows us to encrypt arbitrary TCP connections inside SSL and secure non-SSL aware daemons and protocols by having Stunnel provide the encryption, requiring no changes to the daemon's code. Without it, we cannot implement encryption features on the mail server.

Creating the necessary qmail-pop3d certificate keys:

Here I show you how to set up a self-signed certificate to use with qmail-pop3d, again the principle is the same as for creating a certificate for a Web Server (refer to OpenSSL chapter if you have problem creating the certificates).

Step 1
First you have to know the **F**ully **Q**ualified **D**omain **N**ame (FQDN) of the Central Mail Hub Server for which you want to request a certificate. When your incoming mail server address is smtp.domain.com then the FQDN of your Central Mail Hub Server is smtp.domain.com.

Step 2
Create a self-signed certificate (x509 structure) without a pass-phrase. The req command creates a self-signed certificate when the -x509 switch is used. For certificates signed by commercial **C**ertifying **A**uthority (CA) like Thawte refer to the OpenSSL chapter for the required procedures to follow.

- To create a self-signed certificate, use the following command:

```
[root@deep ssl]# cd /usr/share/ssl
[root@deep ssl]# openssl req -new -x509 -nodes -days 365 -out tmp.pem
Using configuration from /usr/share/ssl/openssl.cnf
Generating a 1024 bit RSA private key
.............................................++++++
...........++++++
writing new private key to 'privkey.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [CA]:
State or Province Name (full name) [Quebec]:
Locality Name (eg, city) [Montreal]:
Organization Name (eg, company) [Open Network Architecture]:
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) [smtp.domain.com]:
Email Address [noc@domain.com]:
```

WARNING: Pay particular attention to the '-nodes' option we have used in the above command, to create the self-signed certificate. The option '-nodes' creates a certificate without a protected pass-phrase, it is very important to create a certificate without a pass-phrase because Qmail server cannot ask you to enter a password before starting its daemon. Also, be sure that you've entered the FQDN (**F**ully **Q**ualified **D**omain **N**ame) of the Central Mail Hub Server when OpenSSL prompts you for the "**Common Name**".

Step 3
Once the self-signed certificate has been created, we must be sure that the future **pop3.pem** file will has both a RSA PRIVATE KEY and a CERTIFICATE section.

- To include the CERTIFICATE section to RSA PRIVATE KEY, use the command:
  ```
  [root@deep ssl]# cat tmp.pem >> privkey.pem
  ```

The above command will include the CERTIFICATE file named "tmp.pem" to the RSA PRIVATE KEY named "privkey.pem".

Step 4
Next, we must place the certificate file in its appropriate directory and rename it "pop3.pem" for qmail-pop3d server to use it.

- To place the file into its appropriate directory, use the following command:
  ```
  [root@deep ssl]# mv privkey.pem certs/pop3.pem
  [root@deep ssl]# chmod 400 certs/pop3.pem
  [root@deep ssl]# chown 0.0 certs/pop3.pem
  [root@deep ssl]# rm -f tmp.pem
  ```

First we move the privkey file, which contains both the RSA PRIVATE KEY and CERTIFICATE section, to the certs directory and rename it pop3.pem for qmail-pop3d to use it for POP protocol. Then we remove the tmp.pem file from our system since it is no longer needed.

Adding the required SSL parameters to the qmail initialization file:

Once the qmail-pop3d certificate has been created and moved to the appropriate location, we have to change the default /etc/init.d/qmail initialization file for qmail-pop3d to be configured to run with SSL support on the server. This is required because qmail-pop3d should use stunnel to start and we have to define some new parameters with stunnel to make the program run with SSL.

Below I show you the lines to add or change in your default qmail initialization file for qmail-pop3d to run with SSL support. Text in bold is what we have added or changed to the default Qmail initialization file.

Step 1
Edit the **qmail** script file (vi /etc/init.d/qmail) and add/change the following lines:

```
#!/bin/bash

# This shell script takes care of starting and stopping Qmail.
#
```

```
# chkconfig: 2345 80 30
# description:  Qmail is a small, fast, secure Mail Transport Agent, which \
#               is the program that moves mail from one machine to another.
#
# processname: qmail-send

# Source function library.
. /etc/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

# If Qmail is not available stop now.
[ -f /usr/bin/qmail-send ] || exit 0

RETVAL=0
prog="Qmail"

start() {
        echo -n $"Starting $prog: "
        qmail-start "`cat /etc/qmail/dot-qmail`" splogger qmail &

        # Here we start qmail-smtpd with AUTH & SSL support.
        stunnel -n smtp -d 25 -o /var/log/maillog -p \
        /usr/share/ssl/certs/smtp.pem -l /usr/bin/tcp-env -- tcp-env \
        /usr/bin/qmail-smtpd /bin/checkpassword /bin/true

        # Here we start qmail-pop3d with AUTH & SSL support.
        stunnel -d 995 -l /usr/bin/qmail-popup `hostname -f` \
        /bin/checkpassword /usr/bin/qmail-pop3d Maildir -p \
        /usr/share/ssl/certs/pop3.pem
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/qmail
        return $RETVAL
}

stop() {
        echo -n $"Shutting down $prog: "
        killproc qmail-send
        killproc stunnel
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/qmail
        return $RETVAL
}

# See how we were called.
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  queue)
        qmail-qstat
        qmail-qread
        ;;
  status)
```

```
        status qmail-send
        RETVAL=$?
        ;;
  restart)
        stop
        start
        RETVAL=$?
        ;;
  condrestart)
        if [ -f /var/lock/subsys/qmail ]; then
                stop
                start
                RETVAL=$?
        fi
        ;;
  *)
        echo $"Usage: $0 {start|stop|queue|status|restart|condrestart}"
        exit 1
esac
exit $RETVAL
```

Step 2
Finally, we have to restart `Qmail` daemon for the changes to take effect.

- To restart `Qmail`, use the following command:

```
[root@deep /]# /etc/init.d/qmail restart
Shutting down Qmail:              [OK]
Starting Qmail:                   [OK]
```

`Qmail` Administrative Tools

The commands listed below are some of the most used, but many more exist. Check the manual pages for more details.

`qmail-showctl`

This command utility allows you to analyze your existing `Qmail` configuration and explains your current setup. It can be useful when you want to verify if modifications made to your existing configuration have been updated by the system.

- To run `qmail-showctl`, use the following command:

```
[root@deep /]# /var/qmail/bin/qmail-showctl
user-ext delimiter: -.
paternalism (in decimal): 2.
silent concurrency limit: 120.
subdirectory split: 23.
user ids: 82, 81, 86, 0, 87, 83, 84, 85.
group ids: 81, 82.

badmailfrom: (Default.) Any MAIL FROM is allowed.
bouncefrom: (Default.) Bounce user name is MAILER-DAEMON.
bouncehost: (Default.) Bounce host name is dev.openna.com.
concurrencylocal: (Default.) Local concurrency is 10.
concurrencyremote: (Default.) Remote concurrency is 20.
databytes: (Default.) SMTP DATA limit is 0 bytes.
defaultdomain: Default domain name is openna.com.
defaulthost: (Default.) Default host name is dev.openna.com.
doublebouncehost: (Default.) 2B recipient host: dev.openna.com.
doublebounceto: (Default.) 2B recipient user: postmaster.
envnoathost: (Default.) Presumed domain name is dev.openna.com.
```

```
helohost: (Default.) SMTP client HELO host name is dev.openna.com.
idhost: Message-ID host name is dev.openna.com.
localiphost: (Default.) Local IP address becomes dev.openna.com.
locals: (Default.) Messages for me are delivered locally.
me: My name is dev.openna.com.
percenthack: (Default.) The percent hack is not allowed.
plusdomain: Plus domain name is openna.com.
qmqpservers:
QMQP server: 207.35.78.4.
queuelifetime: (Default.) Message lifetime in the queue is 604800.
rcpthosts: (Default.) SMTP clients may send messages to any recipient.
morercpthosts: (Default.) No rcpthosts; morercpthosts is irrelevant.
morercpthosts.cdb: (Default.) No effect.
smtpgreeting: (Default.) SMTP greeting: 220 dev.openna.com.
smtproutes: (Default.) No artificial SMTP routes.
timeoutconnect: (Default.) SMTP client connection timeout is 60 seconds.
timeoutremote: (Default.) SMTP client data timeout is 1200 seconds.
timeoutsmtpd: (Default.) SMTP server data timeout is 1200 seconds.
virtualdomains: (Default.) No virtual domains.
```

qmail-qread

This command utility is used to list outgoing messages and recipients on the system in human-readable format. If you want to see your queued messages in the system, then you must use the `qmail-qread` command. `qmail-qread` scans the queue for messages that haven't been completely delivered yet. If a message has multiple recipients, it's not unusual for some of the recipients to receive the message before others.

- To scans the outgoing queue of messages, use the following command:
 `[root@deep /]# qmail-qread`

NOTE: If you want to process `qmail` queues manually, you can send an `ALRM` signal to `qmail-send` daemon to have it run through everything in the queue immediately.
i.e., "`killall -ALRM qmail-send`"

qmail-qstat

The `qmail-qstat` command gives a human-readable breakdown of the number of messages at various stages in the mail queue. To summarize, it summarizes the status of your mail queue.

- To see the status of your mail queue, use the following command:

```
[root@deep /]# qmail-qstat
messages in queue: 0
messages in queue but not yet preprocessed: 0
```

Qmail Users Tools

The commands listed below are some of the most used, but many more exist. Check the manual pages for more details.

maildirwatch

The "`maildirwatch`" program is used to look for new user's mail in a maildir inside terminal screen. This is the program we use to replace the `mailx` package we have uninstalled previously during installation of `Qmail`. Recall that the `maildirwatch` tool is more reliable, fast and secure then `mailx`.

NOTE: If you receive an error message like: `maildirwatch: fatal: MAILDIR not set`

It is because you have forgotten to "give it" the `MAILDIR` variable, for instance:
`export MAILDIR=$HOME/Maildir`

Further documentation

For more details, there are several manual pages about `Qmail` that you could read. I highly recommend you take the time and run through them. By doing this, you'll be more comfortable with the way `Qmail` works.

`$ man bouncesaying (1)`	Bounce each incoming message.
`$ man condredirect (1)`	Redirect mail to another address.
`$ man except (1)`	Reverse the exit code of a program.
`$ man forward (1)`	Forward new mail to one or more addresses.
`$ man maildir2mbox (1)`	Move mail from a maildir to an mbox.
`$ man maildirmake (1)`	Create a maildir for incoming mail.
`$ man maildirwatch (1)`	Look for new mail in a maildir.
`$ man mailsubj (1)`	Send a mail message with a subject line.
`$ man preline (1)`	Prepend lines to message.
`$ man qbiff (1)`	Announce new mail the moment it arrives.
`$ man qreceipt (1)`	Respond to delivery notice requests.
`$ man tcp-env (1)`	Set up TCP-related environment variables.
`$ man addresses (5)`	Formats for Internet mail addresses.
`$ man mbox (5)`	File containing mail messages.
`$ man dot-qmail (5)`	Control the delivery of mail messages.
`$ man envelopes (5)`	Sender/recipient lists attached to messages.
`$ man maildir (5)`	Directory for incoming mail messages.
`$ man qmail-control (5)`	Qmail configuration files.
`$ man qmail-header (5)`	Format of a mail message.
`$ man qmail-log (5)`	The qmail activity record.
`$ man qmail-users (5)`	Assign mail addresses to users.
`$ man tcp-environ (5)`	TCP-related environment variables.
`$ man forgeries (7)`	How easy it is to forge mail.
`$ man qmail (7)`	Overview of qmail documentation.
`$ man qmail-limits (7)`	Artificial limits in the qmail system.
`$ man qmail-newu (8)`	Prepare address assignments for qmail-lspawn.
`$ man qmail-command (8)`	`U`ser-specified mail delivery program.
`$ man qmail-getpw (8)`	Give addresses to users.
`$ man qmail-inject (8)`	Preprocess and send a mail message.
`$ man qmail-local (8)`	Deliver or forward a mail message.
`$ man qmail-lspawn (8)`	`S`chedule local deliveries.
`$ man qmail-newmrh (8)`	Prepare morercpthosts for qmail-smtpd.
`$ man qmail-pop3d (8)`	Distribute mail via POP.
`$ man qmail-popup (8)`	Read a POP username and password.
`$ man qmail-pw2u (8)`	Build address assignments from a passwd file.
`$ man qmail-qmqpc (8)`	Queue a mail message via QMQP.
`$ man qmail-qmqpd (8)`	Receive mail via QMQP.
`$ man qmail-qmtpd (8)`	Receive mail via QMTP.
`$ man qmail-send (8)`	Deliver mail messages from the queue.
`$ man qmail-qread (8)`	List outgoing messages and recipients.
`$ man qmail-qstat (8)`	Summarize status of mail queue.
`$ man qmail-queue (8)`	Queue a mail message for delivery.
`$ man qmail-remote (8)`	Send mail via SMTP.



Part X

Internet Message Access Protocol

CHAPTER 32

tpop3d

IN THIS CHAPTER

1. Compiling - Optimizing & Installing `tpop3d`
2. Configuring `tpop3d`
3. Securing `tpop3d`

Linux tpop3d

Abstract

An **I**nternet **M**essage **A**ccess **P**rotocol server provides access to personal mail and system-wide bulletin boards. It is software that runs in the background and allows users, who use a **M**ail **U**ser **A**gent (MUA) program like Netscape Messenger or MS Outlook to transparently access and read mail on the server. It is important to note that an **I**nternet **M**essage **A**ccess **P**rotocol server is not required on all servers but only on a mail server that runs as a Central Mail Hub Server.

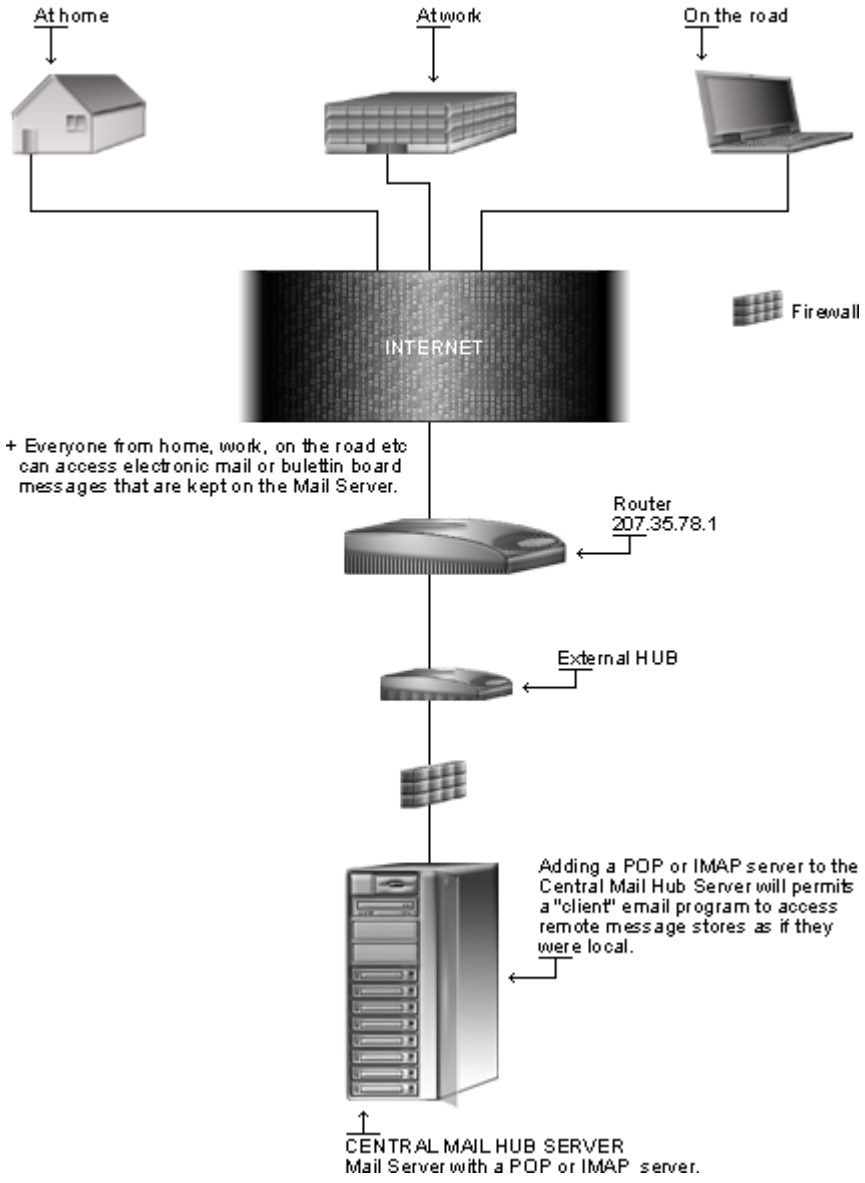
Tpop3d is a fast, extensible, and secure POP3 server. It supports traditional BSD format mailspools (MBOX) and, the Maildir format of Qmail. The main reason why I'm talking about this secure POP software in the book is that it provides support for the Maildir format that we need when we run Exim with Maildir format support on our server. Anyone who installs Exim with Maildir format support really should go with tpop3d as their POP3 server. If you don't want to run Exim with Maildir format support, then you can skip this chapter and choose another type of "Internet Message Access Protocol" program like UW-IMAP or Qpopper.

Maildir format is another method to store mail messages on the server. Its main advantage compared to traditional BSD format mailspools (MBOX) used on many today Unix system is the fact that it provides improved speed to get, delete and read mail messages on highly busy boxes. With Maildir, an MUA can read and delete messages while new mail is being delivered: each message is stored in a separate file with a unique name, so it isn't affected by operations on other messages. An MUA doesn't have to worry about partially delivered mail: each message is safely written to disk in the tmp subdirectory before it is moved to new. The Maildir format is reliable even over NFS.

If you have configured Exim as a Central Mail Hub Server with Maildir format support, you must install tpop3d software or you'll not be able to take advantage of your Linux Mail Server, since Exim is just software that sends mail from one machine to another only. A mail server is a server that is running one or more of the following: an IMAP server, a POP3 server, a POP2 server, or an SMTP server. An example of an SMTP server is Exim that must be already installed on your Linux server as a Central Mail Hub before continuing with this part of the book. Here we are going to cover installing tpop3d.

With tpop3d software, a remote "client" email program can access message stored on the Linux mail server as if they were local. For example, an email is received and stored on a tpop3d server for a user and can be manipulated from his/her computer at home, office, etc, without the need to transfer the messages or files back and forth between these computers.

Internet Message Access Protocol



These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, at personal discretion).
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest tpop3d version number is 1.4.2

The procedures given in this chapter are likely to work on all Linux platforms, but we have only tested it on OpenNA Linux and Red Hat Linux.

Packages

The following is based on information listed by the tpop3d as of 2002/06/24. Please regularly check http://www.ex-parrot.com/~chris/tpop3d/ for the latest status. We chose to install the required component from a source file because it provides the facility to fine tune the installation.

Source code is available from:
Tpop3d Homepage: http://www.ex-parrot.com/~chris/tpop3d/
You must be sure to download: tpop3d-1.4.2.tar.gz

Prerequisites

Tpop3d requires that the software below is already installed on your system to be able to compile successfully. If this is not the case, you must install it. Please make sure you have this program installed on your machine before you proceed with this chapter.

- ✓ ISC BIND & DNS is required to be able to use tpop3d in your system.
- ✓ Exim should be already installed on your system to be able to use tpop3d.

NOTE: For more information on the required software, see their related chapters in this book.

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all the files installed on the system in the eventuality of an update in the future. To solve the problem, it is a good idea to make a list of files on the system before you install tpop3d, and then one afterwards, and then compare them using the **diff** utility to find out what files were placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep root]# find /* > tpop3d1
  ```
- And the following one after you install the software:
  ```
  [root@deep root]# find /* > tpop3d2
  ```
- Then use the following command to get a list of what changed:
  ```
  [root@deep root]# diff tpop3d1 tpop3d2 > Tpop3d-Installed
  ```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. In the example above, we use the `/root` directory of the system to store all generated list files.

Compiling - Optimizing & Installing `tpop3d`

Below are the steps that you must make to configure, compile and optimize the `tpop3d` software before installing it onto your system. First off, we install the program as the user "`root`" so as to avoid permissioning problems.

Step 1
Once you get the program from the main software site you must copy it to the `/var/tmp` directory and change to this location before expanding the archive.

- This can be done with the following commands:

```
[root@deep /]# cp tpop3d-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf tpop3d-version.tar.gz
```

Step 2
After that, move into the newly created `tpop3d` directory and perform the following steps before compiling and optimizing it.

- To move into the newly created `tpop3d` directory, use the following command:

```
[root@deep tmp]# cd tpop3d-1.4.2/
```

Step 3
It is important to set our optimization flags for the compilation of `tpop3d` software on the server to fit our `CPU` architecture on Linux.

- Edit the **`Makefile`** file (`vi +85 Makefile`) and change the line:

```
CFLAGS = -Wall -Wstrict-prototypes -g -O2
```

To read:

```
CFLAGS = -Wall -Wstrict-prototypes -O2 -march=i686 -funroll-loops
```

Step 4
Once the modification has been made to the `tpop3d` source file as shown above, it is time to compile and optimize it for our system.

- To configure and optimize `tpop3d` use the following compilation lines:

```
./configure \
--prefix=/usr \
--sysconfdir=/etc \
--localstatedir=/var \
--mandir=/usr/share/man \
--disable-mbox-bsd \
--enable-mbox-maildir
```

Step 5
At this stage, the program is ready to be built and installed. We build `tpop3d` with the '`make`' command and produce a list of files on the system before we install the software, and one afterwards, then compare them using the **`diff`** utility to find out what files were placed where and finally install `tpop3d`.

```
[root@deep tpop3d-1.4.2]# make
[root@deep tpop3d-1.4.2]# cd
[root@deep root]# find /* > tpop3d1
[root@deep root]# cd /var/tmp/tpop3d-1.4.2/
[root@deep tpop3d-1.4.2]# make install
[root@deep tpop3d-1.4.2]# cd
[root@deep root]# find /* > tpop3d2
[root@deep root]# diff tpop3d1 tpop3d2 > Tpop3d-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

Step 6
Once the compilation, optimization and installation of the software has completed, we can free up some disk space by deleting the program tar archive and the related source directory, since they are no longer needed.

- To delete `tpop3d` and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf tpop3d-version/
[root@deep tmp]# rm -f tpop3d-version.tar.gz
```

Configuring `tpop3d`

After `tpop3d` has been built and installed successfully on your system, the next step is to configure and customize its configuration files to fit your needs.

- ✓ `/etc/tpop3d.conf`: (The `Tpop3d` Configuration File)
- ✓ `/etc/pam.d/tpop3d`: (The `Tpop3d PAM` Support Configuration File)
- ✓ `/etc/init.d/tpop3d`: (The `Tpop3d` Initialization File)

`/etc/tpop3d.conf`: The `Tpop3d` Configuration File

The `/etc/tpop3d.conf` file is the main configuration file for `Tpop3d`. It is in this configuration file that `Tpop3d` gets all of its network information, the name of the server, the domain for which it is responsible, and so forth. A typical `tpop3d.conf` file consists of a number of key: value pairs.

- Create the **`tpop3d.conf`** file (`touch /etc/tpop3d.conf`). Below is what we recommend you set:

```
listen-address:         0.0.0.0
max-children:           1024
log-facility:           mail
mailbox:                maildir:$(home)/Maildir
auth-pam-enable:        yes
auth-pam-mail-group:    mail
```

This tells the `tpop3d.conf` file to set itself up for this particular configuration with:

`listen-address: 0.0.0.0`
The "`listen-address`" directive is used to specify the address on which the `POP` daemon should listen for incoming connections. In our configuration, we use "`0.0.0.0`" to inform the system to listen for connections on any interface on the default port "`110`". Change the above value for your `IP` address. More parameters are available if you want to make it run for virtual hosting. See the manual page of `tpop3d.conf` for more information.

`max-children: 1024`
The "`max-children`" directive is used to specify the maximum number of child processes that may be actively serving connections at any given time. In our configuration, we set this value to "`1024`". Feel free to change it to whatever value you like.

`log-facility: mail`
The "`log-facility`" directive is used to define the "facility" which `tpop3d` should log system log messages to the `/var/log/maillog` file for verification.

`mailbox: maildir:$(home)/Maildir`
The "`mailbox`" directive is one of the most important options in our configuration. It is used to define the location, and optionally the type, of the mailbox to use when a user is authenticated on the system. This is where we inform the `tpop3d` software to run with `Maildir` support for the `POP` protocol. In our configuration, we store mail messages in the users home directory under the `Maildir` directory. This means that you have to ensure that the `Maildir` directory exists in the home directory of the user for which you want to provide `Maildir` format support. This is very important; don't forget to create the `Maildir` directory for the user because the software does not do it automatically for you.

`auth-pam-enable: yes`
The "`auth-pam-enable`" directive is used to enable authentication using **P**luggable **A**uthentication **M**odules with `tpop3d`. Other types of authentication are available, but in our configuration `PAM` is the most secure and only one we need to make the program work.

`auth-pam-mail-group: mail`
The "`auth-pam-mail-group`" directive is used to specify the group name or `GID` under which access to the `Maildir` will take place. In general, you should not change the default setting of "`mail`" if your mailer software (`Exim`) is running with this group name, which should be the case if you have followed what I explaines in the `Exim` chapter of this book.

/etc/pam.d/tpop3d: The Tpop3d PAM Support Configuration File

For increased security of tpop3d, we have compiled it to use the PAM mechanism for password authentication.

Step 1
To be able to use this feature, we must create the /etc/pam.d/tpop3d file and add the following parameters inside it.

- Create the **tpop3d** file (touch /etc/pam.d/tpop3d) and add the following lines:

```
#%PAM-1.0
auth            required        /lib/security/pam_pwdb.so shadow
account         required        /lib/security/pam_pwdb.so
password        required        /lib/security/pam_cracklib.so
password        required        /lib/security/pam_pwdb.so nullok
use_authtok md5 shadow
session         required        /lib/security/pam_pwdb.so
```

Step2
Now, set the permissions of the tpop3d file to be (0640/-rw-r-----) and owned by the super-user 'root' for security reasons.

- To change the permissions and ownership of the tpop3d file, use the commands:

```
[root@deep /]# chmod 640 /etc/pam.d/tpop3d
[root@deep /]# chown 0.0 /etc/pam.d/tpop3d
```

/etc/init.d/tpop3d: The Tpop3d Initialization File

The /etc/init.d/tpop3d script file is responsible for automatically starting and stopping the Tpop3d POP server. Loading the tpop3d daemon as a standalone daemon will eliminate load time and will even reduce swapping since non-library code will be shared.

Please note that the following script is only suitable for Linux operating systems that use SystemV. If your Linux system uses some other method, like BSD, you'll have to adjust the script below to make it work for you.

Step 1
Create the **tpop3d** script file (touch /etc/init.d/tpop3d) and add the following lines:

```
#!/bin/bash

# This shell script takes care of starting and stopping Tpop3d.
#
# chkconfig: 345 50 50
# description: Tpop3d supports the widely used POP3 protocol for downloading \
#              Internet e-mail used by many popular e-mail clients.
#
# processname: tpop3d
# config: /etc/tpop3d.conf
# pidfile: /var/run/tpop3d.pid

# Source function library.
. /etc/init.d/functions

# Source networking configuration.
```

```
. /etc/sysconfig/network

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

# If Tpop3d is not available stop now.
[ -f /usr/sbin/tpop3d ] || exit 0

# Path to the Tpop3d binary.
tpop3d=/usr/sbin/tpop3d

RETVAL=0
prog="Tpop3d"

start() {
        echo -n $"Starting $prog: "
        daemon $tpop3d
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/tpop3d
        return $RETVAL
}

stop() {
        echo -n $"Shutting down $prog: "
        killproc $tpop3d
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/tpop3d
        return $RETVAL
}

# See how we were called.
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  status)
        status $tpop3d
        RETVAL=$?
        ;;
  restart)
        stop
        start
        RETVAL=$?
        ;;
  condrestart)
        if [ -f /var/lock/subsys/tpop3d ]; then
                stop
                start
                RETVAL=$?
        fi
        ;;
  *)
        echo $"Usage: $0 {start|stop|status|restart|condrestart}"
        exit 1
esac
exit $RETVAL
```

Step 2
Once the tpop3d script file has been created, it is important to make it executable, change its default permissions, create the necessary links and then start it. Making this file executable will allow the system to run it, changing its default permissions to allow only the root user to change it for security reasons, and the creation of the symbolic links will let the process control initialization of Linux start the program automatically for you at each system boot.

- To make this script executable and to change its default permissions, use the commands:
```
[root@deep /]# chmod 700 /etc/init.d/tpop3d
[root@deep /]# chown 0.0 /etc/init.d/tpop3d
```

- To create the symbolic rc.d links for Tpop3d, use the following commands:
```
[root@deep /]# chkconfig --add tpop3d
[root@deep /]# chkconfig --level 345 tpop3d on
```

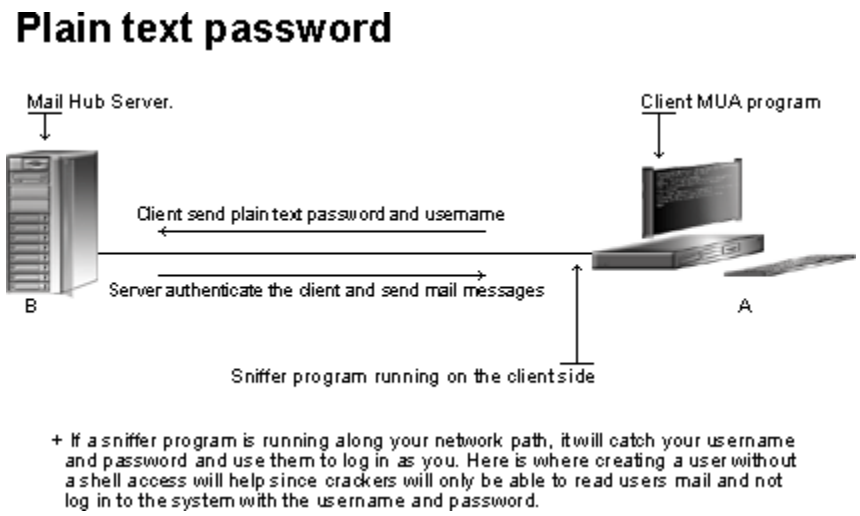
- To start Tpop3d software manually, use the following command:
```
[root@deep /]# /etc/init.d/tpop3d start
Starting Tpop3d:          [OK]
```

Securing tpop3d

This section deals with actions we can take to improve and tighten security under tpop3d. The interesting points here are that we refer to the features available within the base installed program and not to any additional software.

Do you really need tpop3d server and its service?

Be aware that POP program use plain text passwords by default. Anyone running a sniffer program on your network can grab the users username/password and use them to log in. It is not just because you use a POP **M**ail **U**ser **A**gent reader (MUA) like Netscape on your Linux system that you need to run tpop3d server locally. Check your configuration, and if you use a remote/external POP server then uninstall tpop3d from your system. Again, the tpop3d server should ONLY be installed on system running Exim as a Central Mail Hub Server.



The right way to create mail users on the Mail Hub Server:
Just because you have to set up and added a new user to the Mail Hub Server that they user need to have a shell account on the system. Shell accounts are precious and must be given out only if it is necessary. If you only want to allow mail users to get, read and send mails (usually this is what all of us are looking for), then all you have to do is to create a new account for this user without the shell access. Creating a mail user account without shell access on the system will eliminate many risks related to the fact that crackers can use mail user accounts to access the server.

From here, we can give one reason for which having a dedicated machine that runs a Mail Hub Server is important. If you have a server dedicated for electronic mail messages, then the only legitimate user allowed to have login shell access by default to the system, will be the super-user "root". Imagine that you can have, for example, 1000 mail users and even if one of them is compromised, there is no problem since shell access to the system is granted only to our super-user "root".

Step 1
The principle of creating a user without a login shell account is the same as for creating an FTP user without a shell account. This procedure can be applied to any other services for which you want a user without shell access to the system.

- Use the following command to create a new POP user. This step must be done for each additional new user you allow to access your POP server on OpenNA Linux.

```
[root@deep /]# useradd -m -s /bin/false gmourani

[root@deep /]# passwd gmourani
Changing password for user gmourani
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully
```

- Use the following command to create a new POP user. This step must be done for each additional new user you allow to access your POP server on Red Hat Linux.

```
[root@deep /]# useradd -g users -s /bin/false gmourani

[root@deep /]# passwd gmourani
Changing password for user gmourani
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully
```

The above commands should be done for each additional new mail user you allow to access your Central Mail Hub Server. The useradd command will add the new user called "gmourani" to our server. The passwd command will set the password for this user "gmourani".

Step 2
Now, edit the **shells** file (vi /etc/shells) and add a non-existent shell name like "/bin/false", which is the one we used in the passwd command above.

```
[root@deep /]# vi /etc/shells
/bin/bash2
/bin/bash
/bin/sh
/bin/false ← This is our added no-existent shell
```

CHAPTER 33

UW IMAP

IN THIS CHAPTER

1. Compiling - Optimizing & Installing `UW IMAP`
2. Configuring `UW IMAP`
3. Enable `IMAP` or `POP` services via `UCSPI-TCP`
4. Enable `IMAP` or `POP` services via `Xinetd`
5. Securing `UW IMAP`
6. Running `UW IMAP` with `SSL` support

Linux UW IMAP

Abstract
This chapter applies for those who want to run Exim as a Central Mail Hub Server with the IMAP protocol on the traditional Unix MBOX format. If you just want to run Exim as a Central Mail Hub Server with POP protocol on traditional Unix MBOX format, then I recommend you go with Qpopper (see next chapter) instead of UW IMAP because the Qpopper POP protocol is faster.

Imap-2001 from the University of Washington supports SSL client functionality for IMAP & POP3; with this release of the UW IMAP software you don't need any separate SSL modules anymore. If you have configured Exim as a Central Mail Hub Server with native Unix MBOX format (without Maildir format support) and want to use the IMAP protocol to provide email to your users, then you must install the UW IMAP software or you'll not be able to take advantage of your Linux Mail Hub Server, since Exim is software that just sends mail from one machine to another, and nothing else. For now, we are going to cover installing IMAP4, POP3, and POP2, which all come in a single package.

With UW IMAP software, a remote "client" email program can access messages stored on the Linux mail server as if they were local. For example, email received and stored on an IMAP server for a user can be manipulated from his/her computer at home, office, etc, without the need to transfer messages or files back and forth between these computers.

POP stands for "**P**ost **O**ffice **P**rotocol" and simply allows you to list messages, retrieve them, and delete them. IMAP that stands for (**I**nternet **M**essage **A**ccess **P**rotocol) is POP on steroids. It allows you to easily maintain multiple accounts, have multiple people access one account, leave mail on the server, just download the headers, or bodies, no attachments, and so on. IMAP is ideal for anyone on the go, or with serious email needs. The default POP and IMAP servers that most distributions ship fulfill most needs and with the addition of SSL capability UW IMAP becomes now a very powerful, strong and secure program.

Disclaimer
Export Regulations. Software, including technical data, is subject to U.S. export control laws, including the U.S. Export Administration Act and its associated regulations, and may be subject to export or import regulations in other countries. Licensee agrees to comply strictly with all such regulations and acknowledges that it has the responsibility to obtain licenses to export, re-export, or import Software. Software may not be downloaded, or otherwise exported or re-exported (i) into, or to a national or resident of, Cuba, Iraq, Iran, North Korea, Libya, Sudan, Syria or any country to which the U.S. has embargoed goods; or (ii) to anyone on the U.S. Treasury Department's list of Specially Designated Nations or the U.S. Commerce Department's Table of Denial Orders.

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest UW IMAP version number is 2001a

The procedures given in this chapter are likely to work on all Linux platforms, but we have only tested it on OpenNA Linux and Red Hat Linux.

Packages

The following are based on information as listed by UW IMAP as of 2002/06/24. Please regularly check http://www.washington.edu/imap/ for the latest status. We chose to install the required component from source file because it provides the facility to fine tune the installation.

Source code is available from:
UW IMAP Homepage: http://www.washington.edu/imap/
UW IMAP FTP Site: 140.142.3.227
You must be sure to download: imap-2001a.tar.Z

Prerequisites

UW IMAP requires that the software below is already installed on your system to be able to compile successfully. If this is not the case, you must install it. Please make sure you have this program installed on your machine before you proceed with this chapter.

- ✓ OpenSSL is required to run UW IMAP with SSL support on your system.
- ✓ Xinetd is required to be able to use UW IMAP on your system.
- ✓ ISC BIND & DNS is required to be able to use UW IMAP on your system.
- ✓ Exim should be already installed on your system to be able to use UW IMAP.

NOTE: For more information on the required software, see their related chapters in this book.

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all installed files into the system in the eventuality of an updated in the future. To solve the problem, it is a good idea to make a list of files on the system before you install UW IMAP, and one afterwards, and then compares them using the **diff** utility to find out what files are placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep root]# find /* > IMAP1
  ```
- And the following one after you install the software:
  ```
  [root@deep root]# find /* > IMAP2
  ```
- Then use the following command to get a list of what changed:
  ```
  [root@deep root]# diff IMAP1 IMAP2 > IMAP-Installed
  ```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. We use the /root directory of the system to stock all generated list files.

Compiling - Optimizing & Installing UW IMAP

Below are the steps that you must make to configure, compile and optimize the UW IMAP software before installing it into your Linux system. First off, we install the program as user "root" so as to avoid authorization problems.

There are some files we must modify so we can specify the installation paths, compilation and optimizations flags for the Linux system. We must alter those files to be compliant with our Linux file system structure and install/optimize UW IMAP with our PATH Environment variable.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- This can be done with the following commands:
  ```
  [root@deep /]# cp imap-version.tar.Z /var/tmp/
  [root@deep /]# cd /var/tmp/
  [root@deep tmp]# tar xzpf imap-version.tar.Z
  ```

Step 2
After that, move into the newly created UW IMAP directory and perform the following steps before compiling and optimizing it.

- To move into the newly created UW IMAP directory use the following command:
  ```
  [root@deep tmp]# cd imap-2001a/
  ```

Step 3
It is important to set our optimization flags for the compilation of `UW IMAP` software on the server.

- Edit the **`Makefile`** file (`vi +435 src/osdep/unix/Makefile`) and change the line:

  ```
  BASECFLAGS="-g -fno-omit-frame-pointer -O6" \
  ```

 To read:

  ```
  BASECFLAGS="-O2 -march=i686 -funroll-loops" \
  ```

> **NOTE:** You will see many identical or similar lines related to different operating systems in this file. The one, which interests us here, is named "`lnp`" for Linux **P**luggable **A**uthentication **M**odules. It is in this section that we must change the above line. This is important since from release to release this line might change with the addition of new code.

Step 4
Now, we have to change some default installation path to reflect our environment.

- Edit the **`Makefile`** file (`vi +72 src/osdep/unix/Makefile`) and change the lines:

  ```
  ACTIVEFILE=/usr/lib/news/active
  ```

 To read:

  ```
  ACTIVEFILE=/var/lib/news/active
  ```

  ```
  SPOOLDIR=/usr/spool
  ```

 To read:

  ```
  SPOOLDIR=/var/spool
  ```

  ```
  RSHPATH=/usr/ucb/rsh
  ```

 To read:

  ```
  RSHPATH=/usr/bin/rsh
  ```

  ```
  LOCKPGM=/etc/mlock
  ```

 To read:

  ```
  #LOCKPGM=/etc/mlock
  ```

NOTE: The "ACTIVEFILE=" line specifies the path of the "active" directory for UW IMAP, the "SPOOLDIR=" is where the "spool" directory of Linux UW IMAP resides, and the "RSHPATH=" specify the path for "rsh" directory on our system. It's important to note that we don't use rsh services on our server, but even so, we specify the right directory to "rsh".

Step 5
Finally, there are some files to modify to fix some small bugs related to missing headers lines.

- Edit the **news.c** file (vi +24 src/osdep/unix/news.c) and change the line:

  ```
  extern int errno;             /* just in case */
  ```

 To read:

  ```
  #include <time.h>
  ```

- Edit the **phile.c** file (vi +24 src/osdep/unix/phile.c) and change the line:

  ```
  extern int errno;             /* just in case */
  ```

 To read:

  ```
  #include <time.h>
  ```

- Edit the **mh.c** file (vi +24 src/osdep/unix/mh.c) and change the line:

  ```
  extern int errno;             /* just in case */
  ```

 To read:

  ```
  #include <time.h>
  ```

- Edit the **mx.c** file (vi +24 src/osdep/unix/mx.c) and change the line:

  ```
  extern int errno;             /* just in case */
  ```

 To read:

  ```
  #include <time.h>
  ```

Making UW IMAP to compile with SSL support:

If you are interested in compiling UW IMAP to support SSL encryption of usernames and passwords on the IMAP or POP server, then I recommend you follow these steps. If you don't want to compile UW IMAP with SSL support, you can simply skip these steps and go directly to the next section where we will compile the software for our system.

Step 1
The default installation of UW IMAP assumes that OpenSSL, which is required for IMAP/POP with SSL support, has been built under the /usr/local/ssl directory, but because we have a non-standard installation, we must modify the Makefile file to point to the correct locations.

- Edit the **Makefile** file (vi +31 src/osdep/unix/Makefile) and change the lines:

 SSLDIR=/usr/local/ssl

 To read:

 SSLDIR=/usr/share/ssl

 SSLINCLUDE=$(SSLDIR)/include

 To read:

 SSLINCLUDE=$(SSLDIR)/../../include

 SSLLIB=$(SSLDIR)/lib

 To read:

 SSLLIB=$(SSLDIR)/../../lib

Compiling UW IMAP:

Now, we must make a list of all files on the system before installing the software, and one afterwards, then compare them using the **diff** utility to find out what files are placed where and finally we install the UW IMAP software.

Step 1
Compile and install UW IMAP with the following commands.

```
[root@deep imap-2001a]# make lnp SSLTYPE=unix
[root@deep imap-2001a]# cd
[root@deep root]# find /* > IMAP1
[root@deep root]# cd /var/tmp/imap-2001/
[root@deep imap-2001a]# install -m440 ./src/ipopd/ipopd.8c /usr/share/man/man8/ipopd.8c
[root@deep imap-2001a]# install -m440 ./src/imapd/imapd.8c /usr/share/man/man8/imapd.8c
[root@deep imap-2001a]# install -s -m510 ./ipopd/ipop3d /usr/sbin/
[root@deep imap-2001a]# install -s -m510 ./imapd/imapd /usr/sbin/
[root@deep imap-2001a]# install -m444 ./c-client/c-client.a /usr/lib
[root@deep imap-2001a]# ln -s /usr/lib/c-client.a /usr/lib/libc-client.a
[root@deep imap-2001a]# mkdir -p /usr/include/imap
[root@deep imap-2001a]# install -m444 ./c-client/*.h /usr/include/imap/
[root@deep imap-2001a]# install -m444 ./src/osdep/tops-20/shortsym.h /usr/include/imap/
[root@deep imap-2001a]# cd
[root@deep root]# find /* > IMAP2
[root@deep root]# diff IMAP1 IMAP2 > IMAP-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

Note that the **make lnp** command above will configure your Linux system with the **P**luggable **A**uthentication **M**odules (PAM) capability for better password security.

The '**SSLTYPE=unix**' parameter will build UW IMAP with SSL capability enabled. If you don't want to include SSL support with UW IMAP, then all you have to do is to omit the '**SSLTYPE=unix**' parameter in your compile line above, but be aware that you can always run UW IMAP without SSL support even if you have included the '**SSLTYPE=unix**' parameter in your compilation to enable SSL support into the software.

The **mkdir** command will create a new directory named "imap" under /usr/include. This new directory "imap" will keep all header development files related to the imapd program "c-client/*.h", and "shortsym.h" files. The **ln -s** command will create a symbolic link from "c-client.a" file to "libc-client.a" which may be required by some third party programs that you may install in the future.

NOTE: For security reasons, if you only use the imapd service, remove the ipop3d binary from your system. The same applies for ipop3d; if you only use the ipop3d service then remove the imapd binary from your server. If you intend to use both the imapd and ipop3d services then keep both binaries.

Step 2
Once compilation, optimization and installation of the software have been finished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete UW IMAP and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf imap-version/
[root@deep tmp]# rm -f imap-version.tar.Z
```

Configuring UW IMAP

After UW IMAP has been built and installed successfully in your system, your next step is to configure and customize its configuration files to fit your needs.

- ✓ /etc/pam.d/imap: (The IMAP PAM Support Configuration File)
- ✓ /etc/pam.d/pop: (The POP PAM Support Configuration File)

/etc/pam.d/imap: The IMAP PAM Support Configuration File

During the compilation of UW IMAP, we have compiled the software to use **P**luggable **A**uthentication **M**odules (PAM) capability with the '**make lnp**' command.

Step 1
Now, we must configure the software to use PAM password authentication support or it will work. Do to that, you must create the imap file. This PAM file is required only if you intend to provide IMAP service in your system.

- Create the **imap** file (touch /etc/pam.d/imap) and add the following lines:

```
#%PAM-1.0
auth       required     /lib/security/pam_stack.so service=system-auth
account    required     /lib/security/pam_stack.so service=system-auth
```

Step2
Now, set the permissions of the imap file to be (0640/-rw-r-----) and owned by the super-user "root" for security reasons.

- To change the permission mode and ownership of imap file, use:

```
[root@deep /]# chmod 640 /etc/pam.d/imap
[root@deep /]# chown 0.0 /etc/pam.d/imap
```

/etc/pam.d/pop: The POP PAM Support Configuration File

As for the IMAP PAM file above, if you intended use POP instead of IMAP service, you must configure the software to use PAM password authentication support or nothing will work.

Step 1
Once done we need to create the pop file. This PAM file is required only if you intend to provide POP service in your system. If you want to provide IMAP and POP support, then you must create and use the both files (/etc/pam.d/imap and /etc/pam.d/pop).

- Create the **pop** file (touch /etc/pam.d/pop) and add the following lines:

```
#%PAM-1.0
auth       required     /lib/security/pam_stack.so service=system-auth
account    required     /lib/security/pam_stack.so service=system-auth
```

Step2
Now, set the permissions of the pop file to be (0640/-rw-r-----) and owned by the super-user "root" for security reasons.

- To change the permission mode and ownership of pop file, use:

```
[root@deep /]# chmod 640 /etc/pam.d/pop
[root@deep /]# chown 0.0 /etc/pam.d/pop
```

Enable IMAP or POP services via UCSPI-TCP

UCSPI-TCP is super-server software like Xinetd that manages the starting and controlling of software that cannot be run as a daemon on the system. Compared to Xinetd, UCSPI-TCP is faster, less buggy and more secure; therefore I recommend you use it instead of Xinetd to control operation of IMAP & POP servers.

Below I give four different examples, which can be used to start IMAP or POP services depending on your needs with UCSPI-TCP.

Example 1
This section applies only if you want to run IMAP server without SSL connection. Here is the sample command line I recommend you to use to enable the IMAP service (imap):

```
[root@deep /]# tcpserver -c 1024 -DRHl localhost 1.2.3.4 143 /usr/sbin/imapd
```

The above example will run the "/usr/sbin/imapd" binary on port 143 "143" and on IP address 1.2.3.4 with no look-up and TCP_NODELAY "-DRHl localhost" for 1024 "-c 1024" simultaneous connections with tcpserver.

Example 2
This section applies only if you want to run the IMAP server through an SSL connection. Here is the sample command line I recommend you to use to enable the IMAP service with SSL support (imaps):

```
[root@deep /]# tcpserver -c 1024 -DRHl localhost 1.2.3.4 993 /usr/sbin/imapd
```

The above example will run the "/usr/sbin/imapd" binary on port 993 "993" and on IP address 1.2.3.4 with no look-up and TCP_NODELAY "-DRHl localhost" for 1024 "-c 1024" simultaneous connections with tcpserver.

Example 3
This section applies only if you want to run a POP3 server without an SSL connection. Here is the sample command line I recommend you to use to enable the POP3 service (pop3):

```
[root@deep /]# tcpserver -c 1024 -DRHl localhost 1.2.3.4 110 /usr/sbin/ipop3d
```

The above example will run the "/usr/sbin/ipop3d" binary on port 110 "110" and on IP address 1.2.3.4 with no look up and TCP_NODELAY "-DRHl localhost" for 1024 "-c 1024" simultaneous connections with tcpserver.

Example 4
This section applies only if you want to run the POP3 server through an SSL connection. Here is the sample command line I recommend you to use to enable the POP3 service with SSL support (pop3s):

```
[root@deep /]# tcpserver -c 1024 -DRHl localhost 1.2.3.4 995 /usr/sbin/ipop3d
```

The above example will run the "/usr/sbin/ipop3d" binary on port 995 "995" and on IP address 1.2.3.4 with no look up and TCP_NODELAY "-DRHl localhost" for 1024 "-c 1024" simultaneous connections with tcpserver.

Enable IMAP or POP services via Xinetd

For those who really prefer to start and control IMAP/POP services through the use of Xinetd for some unknown reason, here are the procedures to follow. Below are four different examples which can be used to start IMAP or POP services depending of your needs with Xinetd.

Example 1
Here is the sample /etc/xinetd.d/imap entry for IMAP service (imap):

- Create the **imap** file (touch /etc/xinetd.d/imap) and add the following lines. Below is the configuration lines required to enable the imap service:

```
# default: on
# description: The IMAP service allows remote users to access their mail
# using an IMAP client such as Mutt, Pine, fetchmail, or Netscape.
#
service imap
{
        socket_type             = stream
        wait                    = no
        user                    = root
        server                  = /usr/sbin/imapd
        only_from               = 0.0.0.0/0
        no_access               = 207.35.78.10
        instances               = 30
        log_on_success          += DURATION HOST
        log_on_failure          += HOST
        nice                    = -2
        disable                 = no
}
```

Example 2
This section applies only if you want to run the IMAP server through an SSL connection. Here is the sample /etc/xinetd.d/imaps entry for the IMAP service with SSL support (imaps):

- Create the **imaps** file (touch /etc/xinetd.d/imaps) and add the following lines:

```
# default: on
# description: The IMAPS service allows remote users to access their mail
# using an IMAP client with SSL support such as Netscape Communicator.
#
service imaps
{
        socket_type             = stream
        wait                    = no
        user                    = root
        server                  = /usr/sbin/imapd
        only_from               = 0.0.0.0/0
        no_access               = 207.35.78.10
        instances               = 30
        log_on_success          += DURATION HOST
        log_on_failure          += HOST
        nice                    = -2
        disable                 = no
}
```

Example 3
Here is the sample `/etc/xinetd.d/pop3` entry for the `POP3` service (`pop3`):

- Create the **pop3** file (`touch /etc/xinetd.d/pop3`) and add the following lines:

```
# default: on
# description: The POP3 service allows remote users to access their mail
# using an POP3 client such as Netscape Communicator, mutt, or fetchmail.
#
service pop3
{
        socket_type             = stream
        wait                    = no
        user                    = root
        server                  = /usr/sbin/ipop3d
        only_from               = 0.0.0.0/0
        no_access               = 207.35.78.10
        instances               = 30
        log_on_success          += DURATION HOST
        log_on_failure          += HOST
        nice                    = -2
        disable                 = no
}
```

Example 4
This section applies only if you want to run the `POP3` server through an `SSL` connection. Here is the sample `/etc/xinetd.d/pop3s` entry for `POP3` service with `SSL` support (`pop3s`):

- Create the **pop3s** file (`vi /etc/xinetd.d/pop3s`) and add the following lines:

```
# default: on
# description: The POP3S service allows remote users to access their mail
# using an POP3 client with SSL support such as fetchmail.
#
service pop3s
{
        socket_type             = stream
        wait                    = no
        user                    = root
        server                  = /usr/sbin/ipop3d
        only_from               = 0.0.0.0/0
        no_access               = 207.35.78.10
        instances               = 30
        log_on_success          += DURATION HOST
        log_on_failure          += HOST
        nice                    = -2
        disable                 = no
}
```

NOTE: Don't forget to restart `Xinetd` for the changes to take effect.

- To restart `Xinetd`, use the following command:

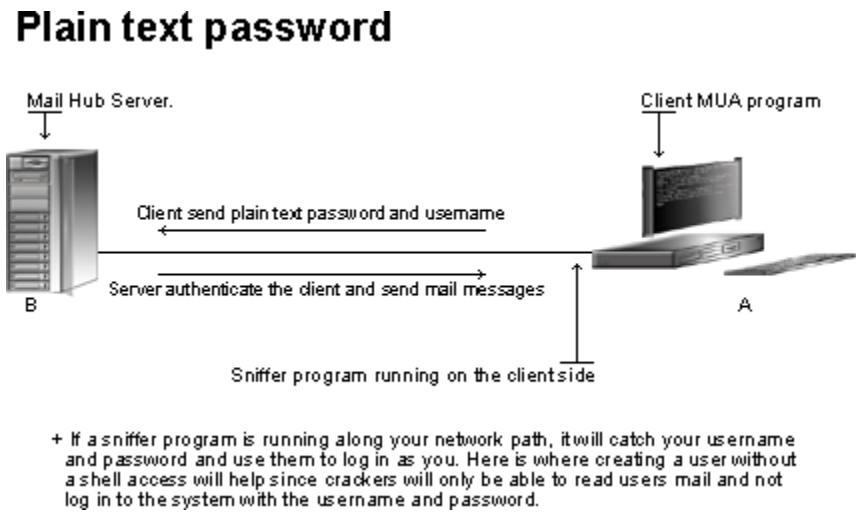
```
[root@deep /]# /etc/init.d/xinetd restart
Stopping Xinetd:                [OK]
Starting Xinetd:                [OK]
```

Securing UW IMAP

This section deals with actions we can take to improve and tighten security under UW IMAP. The interesting points here are that we refer to the features available within the base installed program and not to any additional software.

Do you really need UW IMAP server and its services?

Be aware that IMAP/POP programs use plain text passwords by default. Anyone running a sniffer program on your network path can grab a users username/password and use them to log in as the user. It is not just because you use an IMAP/POP Mail User Agent reader (MUA) like Netscape on your Linux system that you need to run UW IMAP server locally. Check your configuration, and if you use a remote/external IMAP/POP server then uninstall UW IMAP on your system.



The right way to create mail users on the Mail Server:

It is not because you have to set up and add a new user to the Mail Hub Server that this user needs to have a shell account on the system. Shell accounts are precious and must be given out only if it is necessary. If you only want to allow a mail user to get, read and send mails (usually this is what all of us are looking for), then all you have to do is to create a new account for this user without shell access. Creating a mail user account without shell access on the system will eliminate many risks related to the fact that crackers can use a users mail account to access the server.

From here, we can explain one reason why having a dedicated machine that runs a Mail Hub Server is important. If you have a server dedicated just for electronic mail, then the only legitimate user allowed to have login shell access by default to the system will be the super-user "root". Imagine, it this way, you can have, for example, 1000 mail users and even if one of them is compromised, there is no problem, since access to the system can be done only by our super-user "root".

Step 1
The principle of creating a user without a login shell account is the same as for creating an FTP user without a shell account. This procedure can be applied for any other services for which you want a user without shell access to the system.

- Use the following command to create a new UW IMAP user. This step must be done for each additional new user you allow to access your UW IMAP server on OpenNA Linux.

```
[root@deep /]# useradd -m -s /bin/false gmourani

[root@deep /]# passwd gmourani
Changing password for user gmourani
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully
```

- Use the following command to create a new UW IMAP user. This step must be done for each additional new user you allow to access your UW IMAP server on Red Hat Linux.

```
[root@deep /]# useradd -g users -s /bin/false gmourani

[root@deep /]# passwd gmourani
Changing password for user gmourani
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully
```

The above commands should be done for each additional new mail user you allow to access your Central Mail Hub Server. The useradd command will add the new user called "gmourani" to our Linux server. The passwd command will set the password for this user "gmourani".

Step 2
Now, edit the **shells** file (vi /etc/shells) and add a non-existent shell name like "/bin/false", which is the one we used in the passwd command above.

```
[root@deep /]# vi /etc/shells
/bin/bash2
/bin/bash
/bin/sh
/bin/false ← This is our added no-existent shell
```

Running UW IMAP with SSL support

This section applies only if you want to run UW IMAP through an SSL connection. If you are an ISP with many regular users, this may not be the case for you, but if you are a company that provides for your users a mail service, this can be good for you.

We know now that UW IMAP programs use plain text passwords by default. The solution to prevent someone using a sniffer program to grab the username & password of your mail users is to use the new SSL capability of UW IMAP to encrypt the client sessions.

We have already configured UW IMAP, during compilation with SSL support using the special parameter '**SSLTYPE=unix**', therefore UW IMAP is SSL compatible even if you decide to not use its SSL functionality at this time. Now, all we have to do is to set up the certificates.

Below I'll show you how to set up a self-signed certificate to use with UW IMAP, the principle is the same as for creating a certificate for a Web Server (refer to the OpenSSL chapter if you have problems creating the certificates).

Step 1
First you have to know the **F**ully **Q**ualified **D**omain **N**ame (FQDN) of the Central Mail Hub Server for which you want to request a certificate. When your incoming mail server address is smtp.domain.com then the FQDN of your Central Mail Hub Server is smtp.domain.com.

Step 2
Create a self-signed certificate (x509 structure) without a pass-phrase. The req command creates a self-signed certificate when the -x509 switch is used. For certificates signed by commercial **C**ertifying **A**uthority (CA) like Thawte refer to the OpenSSL chapter for the required procedures to follow.

- To create a self-signed certificate, use the following command:

```
[root@deep ssl]# cd /usr/share/ssl
[root@deep ssl]# openssl req -new -x509 -nodes -days 365 -out tmp.pem
Using configuration from /usr/share/ssl/openssl.cnf
Generating a 1024 bit RSA private key
.............................................++++++
...........++++++
writing new private key to 'privkey.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [CA]:
State or Province Name (full name) [Quebec]:
Locality Name (eg, city) [Montreal]:
Organization Name (eg, company) [Open Network Architecture]:
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) [smtp.domain.com]:
Email Address [noc@domain.com]:
```

WARNING: Pay particular attention to the '-nodes' option we have used in the above command, to create the self-signed certificate. The option '-nodes' creates a certificate without a protected pass-phrase, it is very important to create a certificate without a pass-phrase because UW IMAP server cannot ask you to enter a password before starting its daemon. Also, be sure that you've entered the FQDN (**F**ully **Q**ualified **D**omain **N**ame) of the Central Mail Hub Server when OpenSSL prompts you for the "**Common Name**".

Step 3
Once the self-signed certificate has been created, we must be sure that the future **imapd.pem** file will has both a RSA PRIVATE KEY and a CERTIFICATE section.

- To include the CERTIFICATE section to RSA PRIVATE KEY, use the command:
```
[root@deep ssl]# cat tmp.pem >> privkey.pem
```

The above command will include the CERTIFICATE file named "tmp.pem" to the RSA PRIVATE KEY named "privkey.pem".

Step 4
Next, we must place the certificate file to its appropriate directory and rename it "imapd.pem" if you use IMAP protocol or "ipop3d.pem" if you use POP3 protocol for UW IMAP server to recognize it. If you rename the certificate something other than "imapd.pem" or "ipop3d.pem", be aware that the UW IMAP will not recognize it. In the example, below, we rename the certificate "imapd.pem" because we want to run IMAP protocol for our mail users.

- To place the file into its appropriate directory, use the following command:
```
[root@deep ssl]# mv privkey.pem certs/imapd.pem
[root@deep ssl]# chmod 400 certs/imapd.pem
[root@deep ssl]# rm -f tmp.pem
```

First we move the privkey file which contain both the RSA PRIVATE KEY and CERTIFICATE section to the certs directory and rename it imapd.pem for UW IMAP to use it for the IMAP protocol. Then we remove the tmp.pem file from our system since it is no longer needed.

Step 5
Now, it is important to verify if the new imapd.pem certificate file works before connecting with a client MUA program like Netscape to read mail through SSL. Please make sure that the UCSPI-TCP or Xinetd daemon with the imaps value enabled is already running before proceeding with the test.

- To test your new IMAP certificate, use the following command:
```
[root@deep ssl]# openssl
OpenSSL> s_client -host smtp.domain.com -port 993
CONNECTED(00000003)
depth=0 /C=CA/ST=Quebec/L=Montreal/O=Open Network
Architecture/CN=smtp.domain.com/Email=noc@domain.com
verify error:num=18:self signed certificate
verify return:1
depth=0 /C=CA/ST=Quebec/L=Montreal/O=Open Network
Architecture/CN=smtp.domain.com/Email=noc@domain.com
verify return:1
---
Certificate chain
 0 s:/C=CA/ST=Quebec/L=Montreal/O=Open Network
Architecture/CN=smtp.domain.com/Email=noc@domain.com
   i:/C=CA/ST=Quebec/L=Montreal/O=Open Network
Architecture/CN=smtp.domain.com/Email=noc@domain.com
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIDlTCCAv6gAwIBAgIBADANBgkqhkiG9w0BAQQFADCBlDELMAkGA1UEBhMCQ0Ex
DzANBgNVBAgTBlF1ZWJlYzERMA8GA1UEBxMITW9udHJlYWwxIjAgBgNVBAoTGU9w
```

```
ZW4gTmV0d29yayBBcmNoaXRlY3R1cmUxHjAcBgNVBAMTFXVsbHlzZS5tdHRjb25z
ZWlsLmNvbTEdMBsGCSqGSIb3DQEJARYObm9jQG9wZW5uYS5jb20wHhcNMDAxMjE2
MDQ1NjI2WhcNMDIwNzE3MTU1OTU0WjCBlDELMAkGA1UEBhMCQ0ExDzANBgNVBAgT
BlF1ZWJlYzERMA8GA1UEBxMITW9udHJlYWwxIjAgBgNVBAoTGU9wZW4gTmV0d29y
ayBBcmNoaXRlY3R1cmUxHjAcBgNVBAMTFXVsbHlzZS5tdHRjb25zZWlsLmNvbTEd
MBsGCSqGSIb3DQEJARYObm9jQG9wZW5uYS5jb20wgZ8wDQYJKoZIhvcNAQEBBQAD
gY0AMIGJAoGBAM7HC7h/Vxi3ox5nECmd3odhJwGZFdq4tOvbMkknn3F7HAsEpcpJ
OddtZtHNhN3rDn1vYLzuWc0flmG/ry3G5grshsd8JFHp024kRjsdOZSWjoAcT+UE
hD/jF0Wg8L5nRlOuD1RiU9eGqMma7vG80QKGvq/4y5bKUfLYEdHbCTEnAgMBAAGj
gfQwgfEwHQYDVR0OBBYEFLSZEXinVoRgQjKe8pZt6NWWTOFPMIHBBgNVHSMEgbkw
gbaAFLSZEXinVoRgQjKe8pZt6NWWTOFPoYGapIGXMIGUMQswCQYDVQQGEwJDQTEP
MA0GA1UECBMGUXVlYmVjMREwDwYDVQQHEwhNb250cmVhbDEiMCAGA1UEChMZT3Bl
biBOZXR3b3JrIEFyY2hpdGVjdHVyZTEeMBwGA1UEAxMVdWxseXNlLm10dGNvbnNl
aWwuY29tMR0wGwYJKoZIhvcNAQkBFg5ub2NAb3Blbm5hLmNvbYIBADAMBgNVHRME
BTADAQH/MA0GCSqGSIb3DQEBBAUAA4GBAAJC7BzgXPJ2PezOH1R8I9a/xdW36mpp
6YB08P6p1a3oO5NAauf9KW+1bUd7UAM6c61Jyj2g8oL4v9ukx27Z9r2nE4Y4Jubs
HQ1VuZ9zpqbHINcMRlugCUWSqKdTcYoQNL+EXnPefs6+JjCmEiaTMEmn2Ggm7yE3
ef+0J3LXhrzr
-----END CERTIFICATE-----
subject=/C=CA/ST=Quebec/L=Montreal/O=Open Network
Architecture/CN=smtp.domain.com/Email=noc@domain.com
issuer=/C=CA/ST=Quebec/L=Montreal/O=Open Network
Architecture/CN=smtp.domain.com/Email=noc@domain.com
---
No client certificate CA names sent
---
SSL handshake has read 1075 bytes and written 314 bytes
---
New, TLSv1/SSLv3, Cipher is DES-CBC3-SHA
Server public key is 1024 bit
SSL-Session:
    Protocol  : TLSv1
    Cipher    : DES-CBC3-SHA
    Session-ID:
FB1C9CCF4F540CECEF138625549C0391CAC1BBC84A5FDBC37F6AFC4616D785EA
    Session-ID-ctx:
    Master-Key:
AC9E7F536E5E5C7F3CDE76C9590F95894E5BAE3A0EF2A466867D5A7BD57B44327CAE455D4
EBAFFFE10A6C3B2451A7866
    Key-Arg   : None
    Start Time: 976954222
    Timeout   : 300 (sec)
    Verify return code: 0 (ok)
---
* OK [CAPABILITY IMAP4 IMAP4REV1 STARTTLS LOGIN-REFERRALS AUTH=PLAIN
AUTH=LOGIN] smtp.domain.com IMAP4rev1 2000.284 at Sat, 16 Dec 2000
03:10:22 -0500 (EST)
```

If the results look like the one above, then communications from the Central Mail Hub Server to the client machine are encrypted for `imapd` with the `SSL` protocol. Congratulations!

Further documentation

For more details, there are some `UW IMAP` manual pages that you could read:

```
$ man imapd (8)          - Internet Message Access Protocol server.
$ man ipopd (8)          - Post Office Protocol server.
```

CHAPTER 34

Qpopper

IN THIS CHAPTER

1. Compiling - Optimizing & Installing `Qpopper`
2. Configuring `Qpopper`
3. Securing `Qpopper`
4. Running `Qpopper` with `SSL` support

Linux Qpopper

Abstract

This chapter is applicable for those who want to run Exim as a Central Mail Hub Server with the POP protocol on a traditional Unix MBOX format. If you just want to run Exim as a Central Mail Hub Server with the IMAP protocol on a traditional Unix MBOX format, then I recommend you go with UW IMAP (see previous chapter) instead of Qpopper because Qpopper does not support the IMAP protocol but just POP3.

If you have configured Exim as a Central Mail Hub Server with native Unix MBOX format (without Maildir format support) and want to use the POP protocol to provide email to your users, then you must install Qpopper software or you'll not have the advantage of using your Linux Mail Hub Server, since Exim is just software that sends mail from one machine to another, and nothing else. For now, we are going to cover installing POP3.

Qpopper is a server for the POP3 protocol (this allows users to access their mail using any POP3 client). Qpopper supports the latest standards, and includes a large number of optional features, such as **A**uthenticated **P**ost **O**ffice **P**rotocol (APOP), integration with **P**luggable **A**uthentication **M**odules (PAM) and packages such as OpenSSL to provide **T**ransport **L**ayer **S**ecurity/**S**ecure **S**ockets **L**ayer (TLS/SSL) encryption of all traffic to and from the email client. It provides enhanced performance features and easier administration.

In our configuration, we provide information on how to compile and use Qpopper with PAM and SSL support. We will not talk about the APOP protocol because it is not used by all MUA's and conflicts when used or compiled with PAM support from the source code. If you want to use APOP with Qpopper, you cannot use PAM and if you want to use PAM with Qpopper (as we do), you cannot use APOP protocol.

Also, if you compile Qpopper with APOP support, it is down to you to find a MUA capable of working with APOP, to the best of my knowledge Eudora is capable of working with APOP but not Outlook or Netscape.

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest Qpopper version number is 4.0.4

The procedures given in this chapter are likely to work on all Linux platforms, but we have only tested it on OpenNA Linux and Red Hat Linux.

Packages

The following are based on information as listed by Qpopper as of 2002/06/24. Please regularly check http://www.eudora.com/qpopper/ for the latest status. We chose to install the required component from source file because it provides the facility to fine tune the installation.

Source code is available from:
Qpopper Homepage: http://www.eudora.com/qpopper/
You must be sure to download: qpopper-4.0.4.tar.gz

Prerequisites

Qpopper requires that the software below is already installed on your system to be able to compile successfully. If this is not the case, you must install it. Please make sure you have this program installed on your machine before you proceed with this chapter.

- ✓ OpenSSL is required to run Qpopper with SSL support on your system.
- ✓ ISC BIND & DNS is required to be able to use Qpopper in your system.
- ✓ Exim should be already installed on your system to be able to use Qpopper.

NOTE: For more information on the required software, see their related chapters in this book.

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all installed files into the system in the eventuality of an updated in the future. To solve the problem, it is a good idea to make a list of files on the system before you install Qpopper, and one afterwards, and then compares them using the **diff** utility of Linux to find out what files are placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep root]# find /* > Qpopper1
  ```
- And the following one after you install the software:
  ```
  [root@deep root]# find /* > Qpopper2
  ```
- Then use the following command to get a list of what changed:
  ```
  [root@deep root]# diff Qpopper1 Qpopper2 > Qpopper-Installed
  ```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. Related to our example above, we use the `/root` directory of the system to stock all generated list files.

Compiling - Optimizing & Installing `Qpopper`

Below are the required steps that you must make to configure, compile and optimize the `Qpopper` software before installing it into your Linux system. First off, we install the program as user "`root`" so as to avoid authorization problems.

Step 1
Once you get the program from the main software site you must copy it to the `/var/tmp` directory and change to this location before expanding the archive.

- This can be done with the following commands:

```
[root@deep /]# cp qpopper-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf qpopper-version.tar.gz
```

Step 2
After that, move into the newly created `Qpopper` source directory and perform the following steps to configure and optimize the software for your system.

- To move into the newly created `Qpopper` directory use the following command:

```
[root@deep tmp]# cd qpopper-4.0.4/
```

- To configure and optimize `Qpopper` use the following compilation lines:

```
CFLAGS="-O2 -march=i686 -funroll-loops"; export CFLAGS
./configure \
--prefix=/usr \
--sysconfdir=/etc \
--localstatedir=/var \
--mandir=/usr/share/man \
--enable-cache-dir=/var/spool/mail \
--enable-log-login \
--enable-specialauth \
--enable-shy \
--enable-standalone \
--enable-timing \
--enable-uw-kludge \
--enable-servermode \
--enable-fast-update \
--enable-temp-drop-dir=/var/spool/mail \
--disable-old-spool-loc \
--disable-status \
--with-openssl \
--with-pam
```

This tells `Qpopper` to set itself up for this particular configuration setup with:

- Specify the location of the directory for cache files.
- Log successful user authentications.
- Enable secure crypt or shadow passwords.
- Hide `Qpopper` version number.
- Makes a standalone `POP3` daemon instead of using `Xinetd`.
- Report elapsed time for login, init, and cleanup.
- Check for and hide UW 'Folder Internal Data' messages.
- Enable SERVER_MODE.
- Reduce `I/O` during server-mode updates.
- Specify directory for temporary mail drop.
- Don't check for old spools in old location.
- Don't write 'Status' or 'X-UIDL' headers.
- Use `OpenSSL`.
- Use `PAM` authentication.

Step 3
Now, we must make a list of all existing files on the system before installing the software and one afterwards then compare them using the **`diff`** utility to find out what files are placed where and finally install the `Qpopper`:

```
[root@deep qpopper-4.0.4]# make
[root@deep qpopper-4.0.4]# cd
[root@deep root]# find /* > Qpopper1
[root@deep root]# cd /var/tmp/qpopper-4.0.4/
[root@deep qpopper-4.0.4]# make install
[root@deep qpopper-4.0.4]# cd
[root@deep root]# find /* > Qpopper2
[root@deep root]# diff Qpopper1 Qpopper2 > Qpopper-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

Step 4
Once the configuration, optimization, compilation, and installation of the `Qpopper` software have been accomplished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete `Qpopper` and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf qpopper-version/
[root@deep tmp]# rm -f qpopper-version.tar.gz
```

Configuring Qpopper

After Qpopper has been built and installed successfully on your system, your next step is to configure and customize its configuration files to fit your needs.

- ✓ /etc/qpopper.conf: (The Qpopper Configuration File)
- ✓ /etc/pam.d/pop3: (The Qpopper PAM Support Configuration File)
- ✓ /etc/sysconfig/qpopper: (The Qpopper System Configuration File)
- ✓ /etc/init.d/qpopper: (The Qpopper Initialization Configuration File)

/etc/qpopper.conf: The Qpopper Configuration File

The qpopper.conf file is the main configuration file for Qpopper. It is in this configuration file that Qpopper gets all of its information, features to enable, disable, the name and location of different files to use, the domain or server for which it is responsible, and so forth.

- Create the **qpopper.conf** file (touch /etc/qpopper.conf). Below is what we recommend you set:

```
set clear-text-password         = default
set reverse-lookup              = false
set tls-support                 = default
set chunky-writes               = never
```

This tells the qpopper.conf file to set itself up for this particular configuration with:

set clear-text-password = default
The "clear-text-password" option is used to inform Qpopper if clear text passwords should be allowed or not. The value "default" means that clear text passwords are permitted for all users. Other values are available like "never", which means that clear text passwords are never permitted, "always", which means that clear text passwords are always permitted, "local", which means that clear text passwords are permitted on the local (127.*.*.*) loop back interface only, and "tls", which means that clear text passwords are permitted when TLS/SSL has been negotiated for the session. In general, the default value is suitable for all users. Another interesting value is "tls", but should be used only if you provide SSL support.

set reverse-lookup = false
The "reverse-lookup" option is used to enable or disable the reverse lookups on client IP addresses to avoid whatever overhead is incurred by the reverse DNS lookup. If you set this option to "false" as we do, then DNS lookup is disabled. For better performance and to avoid delays during a POP connection, we should set this option to "false" to disable DNS lookups on the server. This is a performance feature.

set tls-support = default
The "tls-support" option is used to specify if we want to enable or disable TLS/SSL support with Qpopper. Possible choices are "default", which means that TLS/SSL is not supported, "alternate-port", which means that TLS/SSL is enable. If you use Qpopper with SSL support, you should set this option to "alternate-port" to enable SSL support. In our configuration TLS/SSL is not enable and we set the option to "default" to inform the system about our choice. You should define this option even if you don't use TLS/SSL on your server.

```
set chunky-writes                = never
```

Qpopper sends network output to clients in small chunks. By default, Qpopper aggregates data to be sent to clients in large chunks. This may be faster or slower, depending on specifics of both the client and server hardware and networking stacks as well as network elements in between (such as routers). Also, some networking stacks do their own aggregation. Under congested network conditions, larger packets increase the incidence of lost packets and thus client or server timeouts, leading to "POP timeout" or "EOF" errors. When TLS/SSL is in effect, smaller packets increase the overhead needed to send data, which may result in worse performance. In our configuration we set the value to "never", which means to never aggregate data into large chunks. This is a performance feature.

/etc/pam.d/pop3: The Qpopper PAM Support Configuration File

For the increased security of Qpopper, we have compiled it to use PAM mechanism for password authentication.

Step 1
To be able to use this feature, we must create the /etc/pam.d/pop3 file and add the following parameters inside it.

- Create the **pop3** file (touch /etc/pam.d/pop3) and add the following lines:

```
#%PAM-1.0
auth            required        /lib/security/pam_pwdb.so shadow
account         required        /lib/security/pam_pwdb.so
password        required        /lib/security/pam_cracklib.so
password        required        /lib/security/pam_pwdb.so nullok
use_authtok md5 shadow
session         required        /lib/security/pam_pwdb.so
```

Step2
Now, set the permissions of the pop3 file to be (0640/-rw-r-----) and owned by the super-user "root" for security reasons.

- To change the permissions and ownership of the pop3 file, use:

```
[root@deep /]# chmod 640 /etc/pam.d/pop3
[root@deep /]# chown 0.0 /etc/pam.d/pop3
```

/etc/sysconfig/qpopper: The Qpopper System Configuration File

The /etc/sysconfig/qpopper file is used to specify Qpopper system configuration information, such as the IP address & port number on which Qpopper should listen, and the location of the its configuration file.

Step 1
By default, the qpopper file do not exist after installation, we have to create it.

- Create the **qpopper** file (touch /etc/sysconfig/qpopper) and add the lines:

```
# The IP address & port number on which the Qpopper daemon will listen
# can be specified here. The default port number is "110", for POP3 with
# SSL support (POP3s), the port number must be "995" instead of "110".
#IPADDR="127.0.0.1:110"

# Where our Qpopper configuration file (qpopper.conf) is located.
OPTIONS="-f /etc/qpopper.conf"
```

Step2
Now, set the permissions of the qpopper file to be (0644/-rw-r--r--) and owned by the super-user 'root' for security reasons.

- To change the permissions and ownership of the qpopper file, use:

```
[root@deep /]# chmod 644 /etc/sysconfig/qpopper
[root@deep /]# chown 0.0 /etc/sysconfig/qpopper
```

/etc/init.d/qpopper: The Qpopper Initialization File

The /etc/init.d/qpopper script file is responsible for automatically starting and stopping the Qpopper POP3 server. Loading the popper daemon as a standalone daemon will eliminate load time and will even reduce swapping since non-library code will be shared.

Please note that the following script is only suitable for Linux operating systems that use SystemV. If your Linux system uses some other method, like BSD, you'll have to adjust the script below to make it work for you.

Step 1
Create the **qpopper** script file (touch /etc/init.d/qpopper) and add the following lines:

```
#!/bin/bash

# This shell script takes care of starting and stopping Qpopper POP3 protocol.
#
# chkconfig: 345 50 50
# description: Qpopper supports the widely used POP3 protocol for downloading \
#              Internet e-mail used by many popular e-mail clients.
#
# processname: popper
# config: /etc/qpopper.conf
# pidfile: /var/run/popper.pid

# Source function library.
. /etc/init.d/functions

# Source networking configuration.
```

```
. /etc/sysconfig/network

# Source for additional options if we have them.
if [ -f /etc/sysconfig/qpopper ] ; then
        . /etc/sysconfig/qpopper
fi

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

# If Qpopper is not available stop now.
[ -f /usr/sbin/popper ] || exit 0

# Path to the Qpopper binary.
popper=/usr/sbin/popper

RETVAL=0
prog="Qpopper"

start() {
        echo -n $"Starting $prog: "
        daemon $popper $IPADDR $OPTIONS
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/popper
        return $RETVAL
}

stop() {
        echo -n $"Shutting down $prog: "
        killproc $popper
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/popper
        return $RETVAL
}

# See how we were called.
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  status)
        status $popper
        RETVAL=$?
        ;;
  restart)
        stop
        start
        RETVAL=$?
        ;;
  condrestart)
        if [ -f /var/lock/subsys/popper ]; then
                stop
                start
                RETVAL=$?
        fi
        ;;
  *)
        echo $"Usage: $0 {start|stop|status|restart|condrestart}"
```

```
        exit 1
esac
exit $RETVAL
```

Step 2
Once the `qpopper` script file has been created, it is important to make it executable, change its default permissions, create the necessary links and then start it. Making this file executable will allow the system to run it, changing its default permission to allow only the root user to change it for security reasons, and the creation of the symbolic links will let the process control initialization of Linux start the program automatically for you at each system boot.

- To make this script executable and to change its default permissions, use the commands:
```
[root@deep /]# chmod 700 /etc/init.d/qpopper
[root@deep /]# chown 0.0 /etc/init.d/qpopper
```

- To create the symbolic `rc.d` links for `Qpopper`, use the following commands:
```
[root@deep /]# chkconfig --add qpopper
[root@deep /]# chkconfig --level 345 qpopper on
```

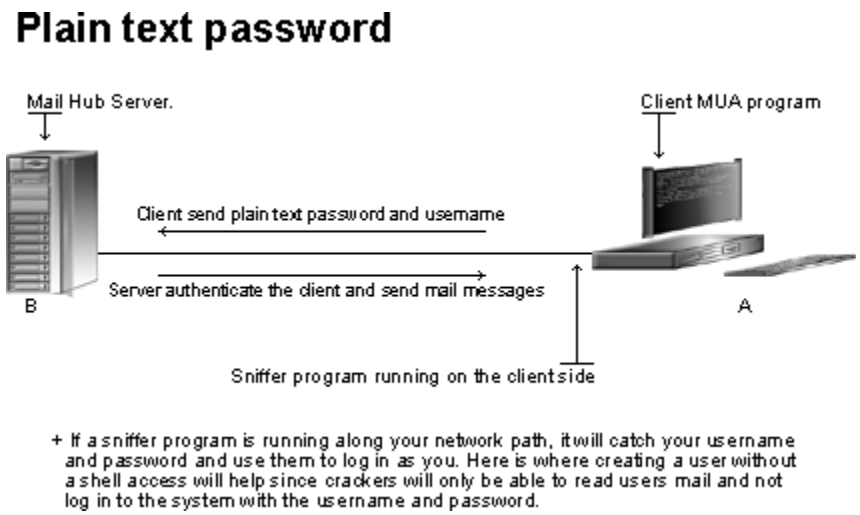
- To start `Exim` software manually, use the following command:
```
[root@deep /]# /etc/init.d/qpopper start
Starting Qpopper:          [OK]
```

Securing `Qpopper`

This section deals with actions we can make to improve and tighten security under `Qpopper`. The interesting points here are that we refer to the features available within the base installed program and not to any additional software.

Do you really need `Qpopper` server and its services?

Be aware that `Qpopper` uses plain text passwords by default. Anyone running a sniffer program on your network path can grab a username & password and use them to log in as a valid user. It is not because you use a `POP` Mail User Agent reader (`MUA`) like `Netscape` on your Linux system that you need to run a `Qpopper` server locally. Check your configuration, and if you use a remote/external `POP` server then uninstall `Qpopper` on your system.



The right way to create mail users on the Mail Server:
It is not because you have to set up and add a new user to the Central Mail Hub Server that this user needs to have a shell account on the system. Shell accounts are precious and must be given out only and only if it is necessary. If you only want to allow mail users to get, read and send mails (usually this is what all of us are looking for), then all you have to do is to create a new account for this user without shell access. Creating a mail user account without shell access to the system will eliminate many risks related to the fact that crackers can use mail user account to access the server.

From here, we can explain one reason for which having a dedicated machine that runs a Central Mail Hub Server is important. If you have a server dedicated for electronic mail, then the only legitimate user allowed to have login shell access by default to the system will be the super-user "root". Imagine, it this way, you can have, for example, 1000 mail users and even if one of them is compromised, there is no problem since access to the system can be done only by our super-user "root".

Step 1
The principle of creating a user without a login shell account is the same as for creating an FTP user without a shell account. This procedure can be applied for any other services for which you want a user without shell access to the system.

- Use the following command to create a new POP user. This step must be done for each additional new user you allow to access your Qpopper server on OpenNA Linux.

```
[root@deep /]# useradd -m -s /bin/false gmourani

[root@deep /]# passwd gmourani
Changing password for user gmourani
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully
```

- Use the following command to create a new POP user. This step must be done for each additional new user you allow to access your Qpopper server on Red Hat Linux.

```
[root@deep /]# useradd -g users -s /bin/false gmourani

[root@deep /]# passwd gmourani
Changing password for user gmourani
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully
```

The above commands should be done for each additional new mail user you allow to access your Central Mail Hub Server. The useradd command will add the new user called "gmourani" to our Linux server. The passwd command will set the password for this user "gmourani".

Step 2
Now, edit the **shells** file (vi /etc/shells) and add a non-existent shell name like "/bin/false", which is the one we used in the passwd command above.

```
[root@deep /]# vi /etc/shells
/bin/bash2
/bin/bash
/bin/sh
/bin/false ← This is our added no-existent shell
```

Running Qpopper with SSL support

This section applies only if you want to run Qpopper through an SSL connection. If you are an ISP with many users, this may not be the case for you, but if you are a company that provides for your users a mail service, this can be good for you.

Qpopper use plain text passwords by default. The solution to prevent someone using a sniffer program to grab the username & password of your mail users is to use the SSL capability of Qpopper to encrypt the client sessions.

We have already configured Qpopper during compilation to enable its SSL support; therefore Qpopper is SSL compatible even if you decide to not use its SSL functionality at this time. Now, all we have to do is to set up the certificates. Below I'll show you how to set up a self-signed certificate to use with Qpopper, and what additional options you must add to the Qpopper configuration file to enable SSL support.

Creating the necessary Qpopper certificate keys:

Below we'll show you how to create a certificate or a self-signed certificate with your own CA certificate for Qpopper. The principle is exactly the same as for creating a certificate or a self-signed certificate for a Web Server. We'll assume that your own CA certificates have been already created, if this is not the case, please refer to OpenSSL chapter for further information.

Step 1
First you have to know the **F**ully **Q**ualified **D**omain **N**ame (FQDN) of the Central Mail Hub Server for which you want to request a certificate. When your incoming mail server address is smtp.domain.com then the FQDN of your Central Mail Hub Server is smtp.domain.com.

Step 2
Create a self-signed certificate (x509 structure) without a pass-phrase. The req command creates a self-signed certificate when the -x509 switch is used. For certificates signed by commercial **C**ertifying **A**uthority (CA) like Thawte refer to the OpenSSL chapter for the procedures to follow.

- To create a self-signed certificate, use the following command:

```
[root@deep ssl]# cd /usr/share/ssl
[root@deep ssl]# openssl req -new -x509 -nodes -days 365 -out tmp.pem
Using configuration from /usr/share/ssl/openssl.cnf
Generating a 1024 bit RSA private key
.....................................++++++
writing new private key to 'privkey.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [CA]:
State or Province Name (full name) [Quebec]:
Locality Name (eg, city) [Montreal]:
Organization Name (eg, company) [Open Network Architecture]:
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) [smtp.domain.com]:
Email Address [noc@domain.com]:
```

WARNING: Pay special attention to the '-nodes' option we have used, in the above command, to create the self-signed certificate. The option '-nodes' creates a certificate without a protected pass-phrase, it is very important to create a certificate without a pass-phrase because Qpopper server cannot ask you to enter a password before starting its daemon. Also, be sure that you've entered the FQDN (**F**ully **Q**ualified **D**omain **N**ame) of the Central Mail Hub Server when OpenSSL prompts you for the "**Common Name**".

Step 3
Once the self-signed certificate has been created, we must be sure that the future **pop3.pem** file will has both a RSA PRIVATE KEY and a CERTIFICATE section.

- To include the CERTIFICATE section to RSA PRIVATE KEY, use the command:
  ```
  [root@deep ssl]# cat tmp.pem >> privkey.pem
  ```

The above command will include the CERTIFICATE file named "tmp.pem" to the RSA PRIVATE KEY named "privkey.pem".

Step 4
Next, we must place the certificate file to its appropriate directory and rename it "pop3.pem" for Qpopper server to recognize it. In the example, below, we rename the certificate "pop3.pem" because we want to run POP protocol for our mail users.

- To place the file into its appropriate directory, use the following command:
  ```
  [root@deep ssl]# mv privkey.pem certs/pop3.pem
  [root@deep ssl]# chmod 400 certs/pop3.pem
  [root@deep ssl]# rm -f tmp.pem
  ```

First we move the privkey file which contain both RSA PRIVATE KEY and CERTIFICATE section to the certs directory and rename it pop3.pem for Qpopper to use it for POP protocol. Then we remove the tmp.pem file from our system since it is no longer needed.

Step 5
Now, it is important to verify if the new pop3.pem certificate file works before connecting with client MUA program like Netscape to read mail through SSL. Please make sure that the Qpopper daemon is already running before proceeding with the test.

- To test your new POP3 certificate, use the following command:
  ```
  [root@deep ssl]# openssl
  OpenSSL> s_client -host smtp.domain.com -port 995
  CONNECTED(00000003)
  depth=0 /C=CA/ST=Quebec/L=Montreal/O=Open Network
  Architecture/CN=smtp.domain.com/Email=noc@domain.com
  verify error:num=18:self signed certificate
  verify return:1
  depth=0 /C=CA/ST=Quebec/L=Montreal/O=Open Network
  Architecture/CN=smtp.domain.com/Email=noc@domain.com
  verify return:1
  ---
  Certificate chain
  ```

```
 0 s:/C=CA/ST=Quebec/L=Montreal/O=Open Network
Architecture/CN=smtp.domain.com/Email=noc@domain.com
   i:/C=CA/ST=Quebec/L=Montreal/O=Open Network
Architecture/CN=smtp.domain.com/Email=noc@domain.com
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIDlTCCAv6gAwIBAgIBADANBgkqhkiG9w0BAQQFADCBlDELMAkGA1UEBhMCQ0Ex
DzANBgNVBAgTBlF1ZWJlYzERMA8GA1UEBxMITW9udHJlYWwxIjAgBgNVBAoTGU9w
ZW4gTmV0d29yayBBcmNoaXRlY3R1cmUxHjAcBgNVBAMTFXVsbHlzZS5tdHRjb25z
ZWlsLmNvbTEdMBsGCSqGSIb3DQEJARYObm9jQG9wZW5uYS5jb20wHhcNMDAxMjE2
MDQ1NjI2WhcNMDIwNzE3MTU1OTU0WjCBlDELMAkGA1UEBhMCQ0ExDzANBgNVBAgT
BlF1ZWJlYzERMA8GA1UEBxMITW9udHJlYWwxIjAgBgNVBAoTGU9wZW4gTmV0d29y
ayBBcmNoaXRlY3R1cmUxHjAcBgNVBAMTFXVsbHlzZS5tdHRjb25zZWlsLmNvbTEd
MBsGCSqGSIb3DQEJARYObm9jQG9wZW5uYS5jb20wgZ8wDQYJKoZIhvcNAQEBBQAD
gY0AMIGJAoGBAM7HC7h/Vxi3ox5nECmd3odhJwGZFdq4tOvbMkknn3F7HAsEpcpJ
OddtZtHNhN3rDn1vYLzuWc0flmG/ry3G5grshsd8JFHp024kRjsdOZSWjoAcT+UE
hD/jF0Wg8L5nRlOuD1RiU9eGqMma7vG80QKGvq/4y5bKUfLYEdHbCTEnAgMBAAGj
gfQwgfEwHQYDVR0OBBYEFLSZEXinVoRgQjKe8pZt6NWWTOFPMIHBBgNVHSMEgbkw
gbaAFLSZEXinVoRgQjKe8pZt6NWWTOFPoYGapIGXMIGUMQswCQYDVQQGEwJDQTEP
MA0GA1UECBMGUXVlYmVjMREwDwYDVQQHEwhNb250cmVhbDEiMCAGA1UEChMZT3Bl
biBOZXR3b3JrIEFyY2hpdGVjdHVyZTEeMBwGA1UEAxMVdWxseXNlLm10dGNvbnNl
aWwuY29tMR0wGwYJKoZIhvcNAQkBFg5ub2NAb3Blbm5hLmNvbYIBADAMBgNVHRME
BTADAQH/MA0GCSqGSIb3DQEBBAUAA4GBAAJC7BzgXPJ2PezOH1R8I9a/xdW36mpp
6YB08P6p1a3oO5NAauf9KW+1bUd7UAM6c61Jyj2g8oL4v9ukx27Z9r2nE4Y4Jubs
HQ1VuZ9zpqbHINcMRlugCUWSqKdTcYoQNL+EXnPefs6+JjCmEiaTMEmn2Ggm7yE3
ef+0J3LXhrzr
-----END CERTIFICATE-----
subject=/C=CA/ST=Quebec/L=Montreal/O=Open Network
Architecture/CN=smtp.domain.com/Email=noc@domain.com
issuer=/C=CA/ST=Quebec/L=Montreal/O=Open Network
Architecture/CN=smtp.domain.com/Email=noc@domain.com
---
No client certificate CA names sent
---
SSL handshake has read 1075 bytes and written 314 bytes
---
New, TLSv1/SSLv3, Cipher is DES-CBC3-SHA
Server public key is 1024 bit
SSL-Session:
    Protocol  : TLSv1
    Cipher    : DES-CBC3-SHA
    Session-ID:
FB1C9CCF4F540CECEF138625549C0391CAC1BBC84A5FDBC37F6AFC4616D785EA
    Session-ID-ctx:
    Master-Key:
AC9E7F536E5E5C7F3CDE76C9590F95894E5BAE3A0EF2A466867D5A7BD57B44327CAE455D4
EBAFFFE10A6C3B2451A7866
    Key-Arg   : None
    Start Time: 976954222
    Timeout   : 300 (sec)
    Verify return code: 0 (ok)
---
* OK [CAPABILITY IMAP4 IMAP4REV1 STARTTLS LOGIN-REFERRALS AUTH=PLAIN
AUTH=LOGIN] smtp.domain.com IMAP4rev1 2000.284 at Sat, 16 Dec 2000
03:10:22 -0500 (EST)
```

If the results look like the one above, then communications from the Central Mail Hub Server to the client machine are encrypted for `POP3` with the `SSL` protocol. Congratulations!

Starting Qpopper on port 995:

For Qpopper to run with SSL support, we must start it to listen on port 995 instead of the default port 110. This is important because client software expects to connect to port 995 with SSL support.

Step 1
To achieve the change, we have to edit our qpopper file under the /etc/sysconfig directory and make the appropriate alteration to the file.

- Edit your **qpopper** file (vi /etc/sysconfig/qpopper), and change the line.

 IPADDR="0.0.0.0:110"

 To read:

 IPADDR="0.0.0.0:995"

Adding the required SSL parameters to the qpopper.conf file:

Once the Qpopper certificate has been created and moved to the appropriate location, we must add some new options to the qpopper.conf file for Qpopper to be configured to run with SSL support on the server.

Step 1
Below we show you the options to add to your default qpopper.conf file for Qpopper to run with SSL support. Text in bold are what we have added or changed to the default Qpopper configuration file.

- Edit your **qpopper.conf** file (vi /etc/qpopper.conf), and add/change the following options inside the file to enable SSL support with Qpopper.

```
set clear-text-password          = tls
set reverse-lookup               = false
set tls-support                  = alternate-port
set tls-server-cert-file         = /usr/share/ssl/certs/pop3.pem
set chunky-writes                = tls
```

This tells the qpopper.conf file to set itself up for this particular configuration with:

set clear-text-password = tls
The "clear-text-password" option is used to inform Qpopper whatever clear text passwords should be allowed or not. With SSL support, we should change the default value to "tls" to inform Qpopper that clear text passwords are only permitted when TLS/SSL has been negotiated for the session.

set tls-support = alternate-port
The "tls-support" option is used to specify if we want to enable or disable TLS/SSL support with Qpopper. Here we set the option to "alternate-port", which means that TLS/SSL is now enabled with Qpopper.

set tls-server-cert-file = /usr/share/ssl/certs/pop3.pem
The "tls-server-cert-file" option is used to specify the file which contains the server's TLS/SSL certificate we have created previously. Qpopper follows the path to get the certificate to use for the POP3S connection.

```
set chunky-writes                = tls
```

Qpopper sends network output to client in small chunks. By default, Qpopper aggregates data to be sent to clients in large chunks. This may be faster or slower, depending on specifics of both the client and server hardware and networking stacks as well as network elements in between (such as routers). Also, some networking stacks do their own aggregation. Under congested network conditions, larger packets increase the incidence of lost packets and thus client or server timeouts, leading to "POP timeout" or "EOF" errors. When TLS/SSL is in effect, smaller packets increase the overhead needed to send data, which may result in worse performance. In our configuration with SSL support, we set the value to "tls". This is a performance feature.

NOTE: Don't forget to restart your POP server for the changes to take effect.

```
[root@deep /]# /etc/init.d/qpopper restart
Shutting down Qpopper:              [OK]
Starting Qpopper:                   [OK]
```

Further documentation

For more details about Qpopper program, there is one manual page that you could read:

```
$ man popper (8)                - POP3 server.
```



Part XI

Anti-Spam & Anti-Virus

CHAPTER 35

SpamAssassin

IN THIS CHAPTER

1. Compiling - Optimizing & Installing `SpamAssassin`
2. Configuring `SpamAssassin`
3. Testing `SpamAssassin`
4. Running `SpamAssassin` with `Exim`
5. Running `SpamAssassin` with `Qmail`

Linux SpamAssassin

Abstract

Junk email (spam) is a significant security problem for computer system administrators and vendors. Spammers steal resources, disrupt operations, and damage systems. Craig Napier, one of my friends, sent me an email about the spammer's problem with its servers. His email clearly describes the problem that most of us encounter; therefore I've decided to include part of his email message below.

With most of the European countries outlawing SPAM, and now Washington and California State having some good laws on the books, SPAMMERS are getting desperate. It actually seems like the last two-three months have gotten MUCH worse in regards to SPAM possibly because of several new virus/worms containing SMTP servers' right in their package.

SPAM by far is the number one problem with running a system and that's truly an understatement. SPAMMERS are the next best thing to a cracker... In my eyes, they are one and the same... and all the poking and prodding that goes on daily, isn't so much crackers looking for a way to get in, as it is SPAMMERS looking for ways to abuse and use a system <Craig Napier>

The spammer credo is, "*Why pay for expensive network and computer resources when we can just steal yours?"*

Do you want to be in the list of the victims? If no, then install SpamAssassin and you will stop them using your servers' resources to pass their garbage messages.

An electronic message is categorized as "spam" if:

1) The recipient's personal identity and context are irrelevant because the message is equally applicable to many other potential recipients;

2) The recipient has not verifiably granted deliberate, explicit, and still-revocable permission for it to be sent;

3) The transmission and reception of the message appears to the recipient to give a disproportionate benefit to the sender.

SpamAssassin is a mail filter that attempts to identify spam using text analysis and several internet-based real-time blacklists. Using its rule base, it uses a wide range of heuristic tests on mail headers and body text to identify "spam", also known as unsolicited commercial email. Once identified, the mail can then be optionally tagged as spam for later filtering using the user's own mail user-agent application.

In its most recent test, SpamAssassin differentiated between spam and non-spam mail correctly in **99.94%** of cases. It requires very little configuration; you don't need to continually update it with details of your mail accounts, mailing list memberships, etc. It accomplishes filtering without this knowledge as much as possible.

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest SpamAssassin version number is 2.31

The procedures given in this chapter are likely to work on all Linux platforms, but we have only tested it on OpenNA Linux and Red Hat Linux.

Packages

Please check http://spamassassin.org/ regularly for the latest status. We chose to install from source because it provides the facility to fine tune the installation.

Source code is available from:
SpamAssassin Homepage: http://spamassassin.org/
You must be sure to download: Mail-SpamAssassin-2.31.tar.gz

Prerequisites

SpamAssassin requires that the listed software below be already installed on your system to be able to compile successfully. If this is not the case, you must install it from your Linux CD-ROM or source archive files. Please make sure you have this program installed on your machine before you proceed with this chapter.

- ✓ An MTA of your choice like Exim or Qmail.
- ✓ Perl, since SpamAssassin highly depends on it to work.

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all the files installed onto the system if you want to update the package in the future. To solve this problem, it's a good idea to make a list of files on the system before you install SpamAssassin, and then one afterwards, and then compare them using the **diff** utility to find out what files were placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep root]# find /* > Spam1
  ```
- And the following one after you install the software:
  ```
  [root@deep root]# find /* > Spam2
  ```
- Then use the following command to get a list of what changed:
  ```
  [root@deep root]# diff Spam1 Spam2 > SpamAssassin-Installed
  ```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. In our example above, we use the /root directory of the system to store all the generated file lists.

Compiling - Optimizing & Installing SpamAssassin

Below are the steps that you must make to configure, compile and optimize the SpamAssassin software before installing it on your system. First off, we install the program as user 'root' so as to avoid any authorization problems.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- This can be done with the following commands:

```
[root@deep /]# cp Mail-SpamAssassin-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf Mail-SpamAssassin-version.tar.gz
```

Step 2
After that, move to the newly created SpamAssassin source directory and perform the following steps to configure and optimize the software for your system.

- To move into the newly created SpamAssassin directory use the following command:

```
[root@deep tmp]# cd Mail-SpamAssassin-2.31/
```

Step 3
Now, we must make a list of all files on the system before installing the software, and one afterwards, then compare them using the **diff** utility to find out what files are placed where and finally we install the SpamAssassin software:

```
[root@deep Mail-SpamAssassin-2.31]# perl Makefile.PL
[root@deep Mail-SpamAssassin-2.31]# make
[root@deep Mail-SpamAssassin-2.31]# make test
[root@deep Mail-SpamAssassin-2.31]# cd
[root@deep root]# find /* > Spam1
[root@deep root]# cd /var/tmp/Mail-SpamAssassin-2.31/
[root@deep Mail-SpamAssassin-2.31]# mkdir -p /etc/mail/spamassassin
[root@deep Mail-SpamAssassin-2.31]# make install
[root@deep Mail-SpamAssassin-2.31]# cd
[root@deep root]# chmod 0444 /usr/share/spamassassin/*
[root@deep root]# chmod 0640 /etc/mail/spamassassin/*
[root@deep Mail-SpamAssassin-2.31]# cd
[root@deep root]# find /* > Spam2
[root@deep root]# diff Spam1 Spam2 > SpamAssassin-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

Step 4
Once the configuration, optimization, compilation, and installation of the SpamAssassin software have been accomplished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete SpamAssassin and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf Mail-SpamAssassin-version/
[root@deep tmp]# rm -f Mail-SpamAssassin-version.tar.gz
```

Configuring SpamAssassin

After SpamAssassin has been built and installed successfully in your system, your next step is to configure and customize its configuration files to fit your needs.

- ✓ /etc/mail/spamassassin/local.cf: (The SpamAssassin Configuration File)
- ✓ /etc/init.d/spamd: (The SpamAssassin Initialization File)

From the default install of this software, there are no configuration files to modify, the default entries look fine and will work for all needs. SpamAssassin is a small Perl program that really doesn't need any adjustment to work on your server.

If you want to make some personal modification to the default entries, all you have to do is to edit the related SpamAssassin configuration files located under /etc/mail/spamassassin and /usr/share/spamassassin directories. For more information about the operation of each one, check the SpamAssassin manual pages.

/etc/init.d/spamd: The SpamAssassin Initialization File

Two different methods of initialization can be used with SpamAssassin. We can use the spamassassin executable perl file or the spamd program, which provides a daemonized version of the spamassassin executable for better performance. The spamd binary is the one we will use since it improves throughput performance for automated mail checking.

The /etc/init.d/spamd script file is responsible to automatically starting and stopping the spamd daemon server on your Linux system. Please note that the following script is suitable for Linux operating systems that use System V. If you Linux system use some other methods like BSD, you'll have to adjust the script bellow to make it work for you.

Step 1
Create the **spamd** script file (touch /etc/init.d/spamd) and add the following lines inside it:

```
#!/bin/bash

# This shell script takes care of starting and stopping SpamAssassin.
#
# chkconfig: 2345 80 30
# description: spamd is a daemon process which uses SpamAssassin to check \
#              email messages for SPAM. It is normally called by spamc \
#              from a MDA.
#
# processname: spamd

# Source function library.
. /etc/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

# If SpamAssassin is not available stop now.
[ -f /usr/bin/spamd ] || exit 0

# Path to the SpamAssassin binary.
spamd=/usr/bin/spamd
```

```
RETVAL=0
prog="Spamd"

start() {
        echo -n $"Starting $prog: "
        daemon $spamd -d -i 0.0.0.0 -x -F0 -u mail
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/spamd
        return $RETVAL
}

stop() {
        echo -n $"Shutting down $prog: "
        killproc $spamd
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/spamd
        return $RETVAL
}

# See how we were called.
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  status)
        status $spamd
        RETVAL=$?
        ;;
  restart)
        stop
        start
        RETVAL=$?
        ;;
  condrestart)
        if [ -f /var/lock/subsys/spamd ]; then
                stop
                start
                RETVAL=$?
        fi
        ;;
  *)
        echo $"Usage: $0 {start|stop|status|restart|condrestart}"
        exit 1
esac
exit $RETVAL
```

WARNING: If you expect to run the above initialization script with `Qmail`, you should absolutely change the following line:

```
daemon $spamd -d -i 0.0.0.0 -x -F0 -u mail
```

To read:

```
daemon $spamd -d -i 0.0.0.0 -x -F0 -u qmaild
```

Step 2
Once the spamd script file has been created, it is important to make it executable, change its default permissions, create the necessary links and start it. Making this file executable will allow the system to run it, changing its default permission is to allow only the root user to change this file for security reasons, and creation of the symbolic links will let the process control initialization start the program automatically for you at each system boot.

- To make this script executable and to change its default permissions, use the commands:
```
[root@deep /]# chmod 700 /etc/init.d/spamd
[root@deep /]# chown 0.0 /etc/init.d/spamd
```

- To create the symbolic rc.d links for SpamAssassin, use the following command:
```
[root@deep /]# chkconfig --add spamd
[root@deep /]# chkconfig --level 2345 spamd on
```

- To start SpamAssassin software manually, use the following command:
```
[root@deep /]# /etc/init.d/spamd start
Starting Spamd:          [OK]
```

Testing SpamAssassin

Once our Anti-Spam software is started, we have to run some recommended tests to make sure SpamAssassin is working correctly on our system. The tests should all complete successfully with no problems or you will eventually lost mail messages. There are two tests to do; the first test is to scan for a mail that isn't spam and the second for a spam mail message. Again both tests should complete successfully.

To be able to successfully make the tests, we have to move to the SpamAssassin source directory where we have installed the software. Inside the source directory, we will find two text files related to these tests. These text files will be used to test SpamAssassin functionality. You don't have to modify the contents of these files, just use them.

- To move to the SpamAssassin source directory use the following command:
```
[root@deep /]# cd /var/tmp/Mail-SpamAssassin-2.31/
```

Test 1 - Scanning a no spam message

In this test, we will scan a no spam mail message to verify if SpamAssassin can detect it. An example text file called "sample-nonspam.txt" is available inside the source directory of the software. We will use it for our test.

- To scan a no spam mail for our first test, use the following command:
```
[root@deep SpamAssassin]# spamassassin -t < sample-nonspam.txt | less
```

Look for a line like this in the output:
```
X-Spam-Status: No, hits=-2.8 required=5.0
tests=GAPPY_TEXT,LINES_OF_YELLING,PGP_SIGNATURE version=2.31
```

Test 2 - Scanning for a bad spam message
In this test, we will scan a bad spam mail message to verify if SpamAssassin can detect it. An example text file called "sample-spam.txt" is available inside the source directory of the software. We will use it for our test.

- To scan for a bad spam mail for our second test, use the following command:
 [root@deep SpamAssassin]# **spamassassin -t < sample-spam.txt | less**

Look for a line like this in the output:

```
X-Spam-Status: Yes, hits=15.0 required=5.0
tests=FROM_HAS_MIXED_NUMS,INVALID_MSGID,INVALID_DATE,MSGID_HAS_NO_AT,SMTPD_IN_RC
VD,UNDISC_RECIPS,NO_REAL_NAME,HOME_EMPLOYMENT,ONCE
_IN_LIFETIME,CALL_FREE,REMOVE_SUBJ,LINES_OF_YELLING,LINES_OF_YELLING_2,LINES_OF_
YELLING_3 version=2.31
```

Running SpamAssassin with Exim

This section applies only if you want to run SpamAssassin with Exim. Remember that SpamAssassin after its default installation cannot automatically filter incoming or outgoing mail messages. You have to do it manually, and this is not what we want to do, therefore we have to integrate it with the MTA software (Exim in our case) we use. In this way, our MTA (Exim) will automatically call SpamAssassin every time it receives or sends mail messages. This allows us to automatically filter mail received and sent via Exim for the entire network.

Necessary steps to integrate SpamAssassin with Exim:

Procedures to allow SpamAssassin to run with Exim are not difficult to accomplish since the majority of the configuration will happen inside the exim.conf file, but we have to be careful of the order in which the additional configuration lines are added to the exim.conf file.

Step 1
First, we have to include new router conditions that will be used for any message that wasn't received from SpamAssassin, wasn't received via a pipe from a local user, and isn't already flagged. This is done by adding the following lines at the top of the "Routers Configuration Section" of exim.conf file. Bold text is what we've added to the default configuration file.

Add the following lines at the TOP of the "**Routers Configuration Section**".

- Edit **exim.conf** file (vi /etc/mail/exim.conf) and add the following lines at the top of the "Routers Configuration Section".

```
begin routers

spamcheck_router:
  no_verify
  check_local_user
  condition = "${if and { {!def:h_X-Spam-Flag:} \
  {!eq {$received_protocol}{spam-scanned}}} {1}{0}}"
  driver = accept
  transport = spamcheck

dnslookup:
  driver = dnslookup
  domains = ! +local_domains
  transport = remote_smtp
  ignore_target_hosts = 0.0.0.0 : 127.0.0.0/8
  no_more
```

Step 2
Second, we have to include the scanning (filtering) by SpamAssassin in the delivery of a message. This is done by adding the following lines at the end of the “Transports Configuration Section” of the exim.conf file.

Add the following lines at the END of the “**Transports Configuration Section**”.

- Edit **exim.conf** file (vi /etc/mail/exim.conf) and add the following lines at the end of the "Transports Configuration Section".

```
address_reply:
  driver = autoreply

spamcheck:
  driver = pipe
  batch_max = 100
  command = /usr/sbin/exim -oMr spam-scanned -bS
  use_bsmtp = true
  transport_filter = /usr/bin/spamc
  home_directory = "/tmp"
  current_directory = "/tmp"
  user = mail
  group = mail
  log_output = true
  return_fail_output = true
  return_path_add = false
  message_prefix =
  message_suffix =
```

Step 3
Now, we have to restart Exim daemon for the changes to take effect.

- To restart Exim, use the following command:

```
[root@deep /]# /etc/init.d/exim restart
Shutting down Exim:              [OK]
Starting Exim:                   [OK]
```

NOTE: Please note that SpamAssassin daemon (spamd) should be already started on your server. If this is not the case, then start it now.

Running SpamAssassin with Qmail

This section applies only if you want to run SpamAssassin with Qmail. Remember that SpamAssassin after its default installation cannot automatically filter incoming or outgoing mail messages. You have to do it manually, and this is not what we want to do, therefore we have to automate the procedure for the entire network. Integrating SpamAssassin with Qmail can do this. This way, our Qmail server will automatically call SpamAssassin every time it receives or sends mail messages for users on our system.

Installing the `safecat` software:

As you should know by now, there is always new software to add with `Qmail` every time you want to provide a new feature. This is the way `Qmail` works and an Anti-Spam features are not exceptions. We have to retrieve and install a small program called "`safecat`". This new software implements Dan Bernstein's maildir algorithm, and can copy standard input safely to a specified directory. With `safecat`, the user is offered two assurances. First, if `safecat` returns successfully, then all data is guaranteed to be saved in the destination directory. Second, if a file exists in the destination directory, placed there by `safecat`, then the file is guaranteed to be complete. `SpamAssassin` requires `safecat` with `Qmail` to be able to delivers bounced messages to users mail directories and without it, `Qmail` will not be able to correctly process mail messages or spam on your server.

Step 1
First, we have to get the program (http://www.nb.net/~lbudney/linux/software/safecat.html) and copy it to the `/var/tmp` directory of our Linux system and change to this location before expanding the archive. Next, we have to move into the newly created `safecat` directory and perform the following steps to compile and optimize it.

```
[root@deep /]# cp safecat-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf safecat-version.tar.gz
[root@deep tmp]# cd safecat-1.8
```

Step 2
Before going into compilation of the program, we'll edit the `conf-cc` file and change the default compiler flags to fit our own `CPU` architecture for better performance.

- Edit the **`conf-cc`** file (`vi conf-cc`) and change the line:

  ```
  cc -O2
  ```

 To read:

  ```
  gcc -O2 -march=i686 -funroll-loops
  ```

Step 3
Also, we have to edit the `conf-root` file and change the default top-level directory for installation of the software to reflect our installation location.

- Edit the **`conf-root`** file (`vi conf-root`) and change the line:

  ```
  /usr/local
  ```

 To read:

  ```
  /usr
  ```

Step 4
Now, we must make a list of files on the system before you install the software, and one afterwards then compare them using the **diff** utility to find out what files are placed where and finally install safecat in the system.

```
[root@deep safecat-1.8]# make
[root@deep safecat-1.8]# cd
[root@deep root]# find /* > Safecat1
[root@deep root]# cd /var/tmp/safecat-1.8/
[root@deep safecat-1.8]# make setup check
[root@deep safecat-1.8]# cd
[root@deep root]# find /* > Safecat2
[root@deep root]# diff Safecat1 Safecat2 > Safecat-Installed
```

Creating a .qmail file for each mail user accounts:

Qmail differs from how Exim manages SpamAssassin, in the way that with Qmail you have to create a ".qmail" file under each users home directory to activate spam protection for the specific user in question. If you don't create the ".qmail" file and populate it with the right parameters under the users home directory, then this user will NOT have spam protection enabled for their account.

Step 1
We have to create a ".qmail" file for each user we want to provide spam protection and inside this file there are some parameters to activate SpamAssassin for the user. In the example below, I create the required ".qmail" file with the appropriate parameters for a fictive user called "gmourani". Again, don't forget to do it for all users you want to provide spam protection.

- To create the .qmail file for user gmourani, use the following commands.
  ```
  [root@deep /]# cd /home/gmourani/
  [root@deep gmourani]# touch .qmail
  ```
- Edit the **.qmail** file (vi .qmail) and add the following line inside it.

  ```
  | /usr/bin/spamc | maildir ./Maildir
  ```

NOTE: The "maildir" part of the line is a small script used by the safecat binary.

Step 2
Now, set the permissions of the `.qmail` file to be (`0644/-rw-r--r--`) and owned by the user '`gmourani`' with the group permissions set to "`users`" for security reasons.

- To change the permissions and ownership of the "`.qmail`" file, use:

```
[root@deep /]# chmod 644 /home/gmourani/.qmail
[root@deep /]# chown gmourani.users /home/gmourani/.qmail
```

At this stage of your work, the program is working and the user `gmourani` is protected and will no longer receive spam in their mail account.

NOTE: Please note that `SpamAssassin` daemon (`spamd`) should be already started on your server. If this is not the case, then start it now.

Further documentation

For more details, there are some manual pages about `SpamAssasin` that you should read:

```
$ man spamassassin (1)                          - Mail filter to identify spam using text analysis.
$ man spamc (1)                                 - Client for spamd.
$ man spamd (1)                                 - Daemonized version of spamassassin.
$ man spamproxyd (1)                            - Mail filter to identify spam using text analysis.
$ man Mail::SpamAssassin (3)                    - Mail::Audit spam detector plugin.
$ man Mail::SpamAssassin::Conf (3)              - SpamAssassin configuration file.
$ man Mail::SpamAssassin::PerMsgStatus (3)      - Per-message status (spam or not-spam).
$ man Mail::SpamAssassin::PersistentAddrList (3) - Persistent address list base class.
$ man Mail::SpamAssassin::SMTP::SmartHost (3)   - A simple smarthost module for Net::SMTP::Server.
```

CHAPTER 36

Sophos

IN THIS CHAPTER

1. Compiling & Installing `Sophos`
2. Configuring `Sophos`
3. Testing `Sophos`

Linux Sophos

Abstract

Probably everyone who uses a computer has, at least once, has gotten a virus via an email message or the Internet. If you have never received a virus, then you are very lucky. Contrary to most other operating systems available, Linux is practically immunized against all kinds of viruses that we find in other operating systems. This is possible because on Linux all files and programs have a special permission to run on the OS and only the super-user "root" can do what he wants. Therefore, if somebody sends you a virus, the virus will never be able to execute on your Linux system because it need to have "root" privileges to do it. This is one of the big reasons why Linux cannot be infected by viruses.

Therefore, someone may say; why we need to install Anti-Virus software on Linux?

Because if you install Linux to run as a Central Mail Hub Server in your network, your Linux system becomes a mail server to everybody allowed to connect to it, to send or receive mail, and this is where Anti-Virus software is required. Because, even if Linux is immunized against all kinds of viruses, your workstations, running Windows or MacOS, are not. Do you know that over **90%** of the viruses arrive via email!

When we install an Anti-Virus program on Linux running as a mail server, our mail server will scan and check for all kind of viruses before sending the email message to the final recipient. If a virus is detected in the email, a warning message will be automatically sent to you and the sender of the message to inform them about a virus in their email and you will never receive the virus in question.

Now before going into the installation of Linux Anti-Virus software, I would like to explain something very important. You should consider Anti-Virus software in two distinct parts. An Anti-Virus scanner software and an Anti-Virus interface software. What's the difference?

Anti-Virus scanner software
An Anti-Virus scanner provides virus checking, automatic reporting and disinfection. This means that an Anti-Virus scanner must have a complete database of all known viruses to date and use its database to scan your system, or email messages in our case, to detect possible viruses.

Anti-Virus interface software
An Anti-Virus interface is, in most cases, a script that interfaces a **M**ail **T**ransport **A**gent (MTA) with one or more virus scanners. It is not an Anti-Virus scanner; it's just an "interface" for virus scanning at the eMail gateway in combination with supported Anti-Virus products for Linux.

To be able to provide to our Linux Central Mail Hub Server Anti-Virus features, we have to install Anti-Virus scanner software and an Anti-Virus interface of our choice. Both are required or nothing will work, because each one depends on the other to work on a mail server.

In the following chapters, we will show you how to install and run the software. To begin our implementation, we will talk about Sophos, which is the Anti-Virus scanner, and finally talk about AMaViS, which is the Anti-Virus interface we will use. I choose to go with AMaViS because it is powerful complete and compatible with Exim and Qmail.

Commercial License

Sophos Anti-Virus is NOT free software. You can get a full working evaluation version for testing but you will need a license if you want to permanently use it on your Linux server. The evaluation period is valid for thee months and after this period the software will be disabled. If you want to buy Sophos, you have to contact Sophos at their website (www.sophos.com). Installation of Sophos is a little different from other software installations because of its commercial nature. This means that source code is not available and we have to follow the installation procedure as explained by the vendor to install it on our system.

Again, I repeat, this product is not free software.
After expiration of the evaluation period, you must order this product through:
http://www.sophos.com/products/howtobuy/order.html

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest Sophos version number is 3.58

The procedures given in this chapter are likely to work on all Linux platforms, but we have only tested it on OpenNA Linux and Red Hat Linux.

Packages

The following is based on information listed by Sophos as of 2002/06/01. Please check http://www.sophos.com/ regularly for the latest status. We chose to install from source because it provides the facility to fine tune the installation.

Software is available from:
Sophos Homepage: http://www.sophos.com/
You must be sure to download: linux.intel.libc6.tar.Z

Prerequisites

Sophos requires that the listed software below be already installed on your system to be able to work successfully. If this is not the case, you must install it from your Linux CD-ROM or archive source files. Please make sure you have this program installed on your machine before you proceed with this chapter.

- ✓ An MTA of your choice like Exim or Qmail.
- ✓ Wget, to be able to retrieve updated database of new virus definition via the Internet.

Compiling & Installing Sophos

Below are the steps that you must make to configure, and compile the Sophos software before installing it on your system. First off, we install the program as user 'root' so as to avoid any permission problems.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- This can be done with the following commands:

```
[root@deep /]# cp linux.intel.libc6.tar.Z /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf linux.intel.libc6.tar.Z
```

Step 2
After that, move into the newly created Sophos directory and perform the following steps to configure and compile the software for your system.

- To move into the newly created Sophos directory use the following command:

```
[root@deep tmp]# cd sav-install/
```

- To configure and compile the software for your system, use the following commands:

```
MANPATH=/usr/share/man
export MANPATH
./install.sh \
        -d /usr \
        -m /usr/share/man \
        -s /usr/lib/sophos \
        -ni
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

Step 3
Once the compilation and installation of Sophos have been accomplished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete Sophos and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf sav-install/
[root@deep tmp]# rm -f linux.intel.libc6.tar.Z
```

The rm command as used above will remove all the source files we have used to compile and install Sophos. It will also remove the Sophos compressed archive from the /var/tmp directory.

Configuring Sophos

After Sophos has been built and installed successfully in your system, your next step is to configure and customize its configuration files to fit your needs.

- ✓ /etc/sav.conf: (The Sophos Configuration File)
- ✓ /etc/cron.daily/sophos.cron: (The Sophos Cron File)

/etc/sav.conf: The Sophos Configuration File

The /etc/sav.conf file is the only configuration file of Sophos. It is automatically created for you during installation of the software and has a line "SAV virus data directory", which tells Sophos Anti-Virus where to find the virus data file vdl.dat on your system. By default, it should have the correct location defined, but to be sure, you can edit the file and verify if the following line is correctly defined.

- Edit the **sav.conf** file (vi /etc/sav.conf) and verify if the following line is defined:

```
SAV virus data directory = /usr/lib/sophos
```

/etc/cron.daily/sophos.cron: The Sophos Cron File

The sophos.cron file is a small script executed automatically by the crond program of your server each day to automate the downloading of new IDE files. This allows us to retrieve new virus definition from the Sophos servers and install them to our system to keep it up to date with latest virus signature databases definition. In this way, system administrator can always be sure he has the latest updates without having to search on the Web.

Step 1
Create the **sophos.cron** file (touch /etc/cron.daily/sophos.cron) and add the following lines to it:

```
#!/bin/sh
# Script for Sophos Sweep by Reiner Keller
#
# For Sophos, see also http://www.sophos.com/support/faqs/autodown.html
# ("How to automate the downloading of IDE files").

cd /usr/lib/sophos

/usr/bin/wget -q -N `/usr/bin/sweep -v |/bin/grep "Product version" |/bin/sed -
e "s/.*: \(.\)\.\(..\)$/
http:\/\/www.sophos.com\/downloads\/ide\/\1\2_ides.zip/"`
/usr/bin/unzip -q -n "???_ides.zip"
rm -f *_ides.zip

chmod 644 *
```

Step2
Now, set the permission mode of the sophos.cron file to be (0500/-r-x------) and owned by the super-user 'root' for security reason.

- To change the permission mode and ownership of the sophos.cron file, use:

```
[root@deep /]# chmod 500 /etc/cron.daily/sophos.cron
[root@deep /]# chown 0.0 /etc/cron.daily/sophos.cron
```

Testing Sophos

Once our Anti-Virus scanner software is installed, we have to run a test to make sure Sophos is working correctly on our system. The test should complete successfully without any problems or you will eventually have problems when you scan email messages for possible virus infection. The test is to scan our /usr directory locally on Linux to see if a virus is detected.

As you can see, we can use Sophos to scan our Linux directories for possible viruses. This is useful when we import or backup Microsoft files on Linux (i.e. with Samba), but of course this is not what we intended to do. We run this internal test just to see if the sweep binary of Sophos is capable of running without any errors on our system.

Test 1 - Scanning the /usr directory for possible virus infection

In this test, we will scan our /usr directory to verify if Sophos run correctly on the system.

- To scan the /usr directory, use the following command:
 `[root@deep /]# sweep -f -all -archive /usr`

You should get something like the following output:

```
SWEEP virus detection utility
Version 3.58, June 2002 [Linux/Intel]
Includes detection for 73553 viruses, trojans and worms
Copyright (c) 1989,2002 Sophos Plc, www.sophos.com

System time 10:39:01, System date 13 June 2002
Command line qualifiers are: -f -all -archive

Full Sweeping


89232 files swept in 26 seconds.
No viruses were discovered.
End of Sweep.
```

Further documentation

For more details, there is one manual page about Sophos that you should read:

```
$ man sweep (1)                    - Virus detection and disinfection utility.
```

CHAPTER 37

AMaViS

IN THIS CHAPTER

1. Verifying & installing all the additional prerequisites to run `AMaViS`
2. Compiling - Optimizing & Installing `AMaViS`
3. Running `AMaViS` with `Exim`
4. Running `AMaViS` with `Qmail`
5. Testing `AMaViS`

Linux AMaViS

Abstract

Virus infections often cause big financial losses due to network disruptions, decreased productivity, corrupted data and leaks of confidential data. Also, the company reputation can be in danger if it spreads viruses to its business associates.

As we know now, to be able to implement Anti-Virus features on a mail server, we have to install an Anti-Virus scanner program and an Anti-Virus interface program. The Anti-Virus scanner is responsible for scanning mail messages for viruses and the Anti-Virus interface provides the bridge between the scanner program and the mail server. Without an Anti-Virus interface program, our mail server will never know that we want it to use the Anti-Virus scanner to scan incoming or outgoing mail messages for possible viruses.

AMaViS is software that ensures attachments coming via email are scanned for viruses before they reach a system that they are able to infect. It resides on the server that handles your incoming and outgoing mails. When a mail arrives, or is sent, instead of being delivered directly, is parsed through a script that extracts all attachments from the mail, unpacks (if needed) and scans them using a professional virus scanner program (Sophos).

Remember that to be able to use AMaViS, you have to install Sophos first as described in the previous chapter. Without Sophos, AMaViS will simply not work on your mail server. This chapter of the book is not difficult to implement, but you must be aware that we need to install many external Perl modules and binary programs to make AMaViS work. These are required because AMaViS should have all the possible tools available to be able to compress, uncompress, read, scan, etc all mail messages.

Contrary to other available Anti-Virus interfaces, AMaViS has the power to completely inspect incoming and outgoing mail messages for possible virus infection even if the virus is attached to the message using many different techniques. When properly installed, it will completely protect your network for all kind of known viruses presently available on the Internet. It is down to your Anti-Virus scanner program to be updated to the latest known virus definitions to detect and quarantine viruses and not the job of AMaViS.

As I said before, some external Perl modules are required as well as some external programs (most are related to compression and un-compression of different file formats under UNIX and Windows). Most of the Perl modules and programs are available with your OS distribution, you can install them from your Linux CR-ROM or follow the instructions that I provide in this chapter to install them. In general, all good Linux vendors should provide this software in their Linux distribution.

At the end of this chapter, I also show you how to make AMaViS work with Exim or Qmail. If you want to make it work with other mail software, you will need to research it yourself.

These installation instructions assume

Commands are Unix-compatible.
The source path `is /var/tmp` (note that other paths are possible, as personal discretion).
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account "`root`".
Whether kernel recompilation may be required: No
Latest `AMaViS` version number is `11`

The procedures given in this chapter are likely to work on all Linux platforms, but we have only tested it on OpenNA Linux and Red Hat Linux.

Packages

The following is based on information listed by `AMaViS` as of 2001/07/04. Please check http://www.amavis.org/ regularly for the latest status. We chose to install from source because it provides the facility to fine tune the installation.

Software is available from:
`AMaViS` Homepage: http://www.amavis.org/
You must be sure to download: `amavis-perl-11.tar.gz`

Prerequisites

`AMaViS` requires that the listed software below be already installed on your system to be able to work and install successfully. If this is not the case, you must install them from your Linux CD-ROM or source archive files. Please make sure you have these programs installed on your machine before you proceed with this chapter.

- ✓ An `MTA` of your choice like `Exim` or `Qmail`.
- ✓ `Sophos`, to be able to scan mail messages on your mail server.
- ✓ `Perl` is needed for `AMaViS` and should be already installed on your system.
- ✓ `Shareutils` is needed for `AMaViS` and should be already installed on your system.
- ✓ `Unzip` is needed for `AMaViS` and should be already installed on your system.
- ✓ `Zip` is needed for `AMaViS` and should be already installed on your system.
- ✓ `Ncompress` is needed for `AMaViS` and should be already installed on your system.

Verifying & installing all the additional prerequisites to run `AMaViS`

It is highly recommended to install all of the prerequisite `Perl` modules and compression software that are needed for `AMaViS` to properly run on your system. Below I show how to retrieve and install all the required `Perl` modules and compression packages. We begin our installation with all the required `Perl` modules packages to install and then finish with the compression packages before compiling, optimizing and installing `AMaViS`.

The Perl modules Locations:

The first thing to do will be to retrieve all the prerequisites Perl archive modules from the Internet and the best place to download most of them will be from the CPAN (**C**omprehensive **P**erl **A**rchive **N**etwork) website located at: http://www.cpan.org/.

As usual, the following is based on information listed by CPAN. Please regularly check at http://www.cpan.org/ for the latest status.

CPAN Homepage: http://www.cpan.org/
You must be sure to download: Compress-Zlib-1.16.tar.gz
You must be sure to download: Archive-Tar-0.22.tar.gz
You must be sure to download: Archive-Zip-1.01.tar.gz
You must be sure to download: IO-stringy-2.108.tar.gz
You must be sure to download: libnet-1.11.tar.gz
You must be sure to download: MailTools-1.45.tar.gz
You must be sure to download: MIME-Base64-2.12.tar.gz
You must be sure to download: MIME-tools-5.411a.tar.gz
You must be sure to download: Unix-Syslog-0.98.tar.gz
You must be sure to download: Convert-TNEF-0.17.tar.gz
You must be sure to download: Convert-UUlib-0.212.tar.gz

Installing all the required Perl modules archives:

Once you get the programs from the related websites you must copy them to the /var/tmp directory and change to this location before expanding the archives. It look like there is lot stuff to install but don't be afraid, all install easily and in the same manner without error if you follow the order of installation as shown below.

With Perl program, there is similarity with the installation of between all the module archives, the installation uses the same procedures, we type "perl Makefile.PL" to generate the "Makefile" then "make" to make the installable files, "make all test" to test the new files for any errors before installation and finally "make install" to install the modules and programs into the configured Perl library and binary directories. Note that this procedure will be the same for all Perl modules archive we need to install. This is the way Perl installs programs under *NIX systems.

Step 1
Ok let's go; the first Perl module to install will be Compress-Zlib. This small Perl program will provide a Perl interface to part of the info-zip zlib compression library for AMaViS.

Before installing the program, we have to make some modifications to the source code to fix some bugs and make it to install to the appropriate location on our server.

- Edit the **config.in** file (vi +28 config.in) and change the lines:

  ```
  INCLUDE        = /usr/local/include
  ```

 To read:

  ```
  INCLUDE        = /usr/include
  ```

```
LIB     = /usr/local/lib
```

To read:

```
LIB     = /usr/lib
```

- Edit the **Zlib.xs** file (vi +416 Zlib.xs) and change the line:

```
SvGROW(buf, SIZE) ;
```

To read:

```
SvGROW(buf, SIZE + 1) ;
```

- Edit the **Zlib.xs** file (vi +628 Zlib.xs) and change the line:

```
SvGROW(output, outsize + s->bufsize) ;
```

To read:

```
SvGROW(output, outsize + s->bufsize + 1) ;
```

- Edit the **Zlib.xs** file (vi +681 Zlib.xs) and change the line:

```
SvGROW(output, outsize + s->bufsize) ;
```

To read:

```
SvGROW(output, outsize + s->bufsize + 1) ;
```

- Edit the **Zlib.xs** file (vi +757 Zlib.xs) and change the line:

```
SvGROW(output, outsize + s->bufsize+1) ;
```

To read:

```
SvGROW(output, outsize + s->bufsize + 1) ;
```

```
[root@deep Compress-Zlib-1.16]# perl Makefile.PL
[root@deep Compress-Zlib-1.16]# make
[root@deep Compress-Zlib-1.16]# make all test
[root@deep Compress-Zlib-1.16]# make install
```

Step 2
The second Perl archive will be Archive-Tar. This small Perl program is a Perl module for the creation and manipulation of tar files with AMaViS.

```
[root@deep Archive-Tar-0.22]# perl Makefile.PL
[root@deep Archive-Tar-0.22]# make
[root@deep Archive-Tar-0.22]# make all test
[root@deep Archive-Tar-0.22]# make install
```

Step 3
The `Archive-Zip` module allows `AMaViS` to create, manipulate, read, and write Zip archive files. Zip archives can be created, or you can read from existing zip files. Once created, they can be written to files, streams, or strings.

```
[root@deep Archive-Zip-1.01]# perl Makefile.PL
[root@deep Archive-Zip-1.01]# make
[root@deep Archive-Zip-1.01]# make all test
[root@deep Archive-Zip-1.01]# make install
```

Step 4
`IO-stringy` primarily provides modules to `AMaViS` for performing both traditional and object-oriented on things *other* than normal file handles.

```
[root@deep IO-stringy-2.108]# perl Makefile.PL
[root@deep IO-stringy-2.108]# make
[root@deep IO-stringy-2.108]# make all test
[root@deep IO-stringy-2.108]# make install
```

Step 5
`libnet` is a collection of `Perl` modules which provides a simple and consistent programming interface (`API`) to the client side of various protocols used in the Internet community.

Before installing the program, we have to make one modification to the source code of the software to change the way we want to install it.

- Edit the **`Makefile.PL`** file (`vi +51 Makefile.PL`) and change the line:

  ```
  system(($^O eq 'VMS' ? 'mcr ': ()),$^X, 'Configure')
  ```

 To read:

  ```
  system(($^O eq 'VMS' ? 'mcr ': ()),$^X, 'Configure', '-d')
  ```

```
[root@deep libnet-1.11]# perl Makefile.PL
[root@deep libnet-1.11]# make
[root@deep libnet-1.11]# make all test
[root@deep libnet-1.11]# make install
```

Step 6
`MailTools` is a toolkit that provides a set of `Perl` modules related to mail applications.

```
[root@deep MailTools-1.45]# perl Makefile.PL
[root@deep MailTools-1.45]# make
[root@deep MailTools-1.45]# make all test
[root@deep MailTools-1.45]# make install
```

Step 7
MIME-Base64 contains a base64 encoder/decoder and a quoted-printable encoder/decoder. These encoding methods are specified in RFC 2045 - MIME (**M**ultipurpose **I**nternet **M**ail **E**xtensions). The Base64 encoding is designed to represent arbitrary sequences of octets in a form that need not be humanly readable.

```
[root@deep MIME-Base64-2.12]# perl Makefile.PL
[root@deep MIME-Base64-2.12]# make
[root@deep MIME-Base64-2.12]# make all test
[root@deep MIME-Base64-2.12]# make install
```

Step 8
MIME-tools is a collection of Perl MIME modules for parsing, decoding, and generating single or multipart (even nested multipart) MIME messages.

```
[root@deep MIME-tools-5.411a]# perl Makefile.PL
[root@deep MIME-tools-5.411a]# make
[root@deep MIME-tools-5.411a]# make all test
[root@deep MIME-tools-5.411a]# make install
```

Step 9
Unix-Syslog provides access to the system logger available on most UNIX system via Perl's XSUBs (Perl's C interface).

```
[root@deep Unix-Syslog-0.98]# perl Makefile.PL
[root@deep Unix-Syslog-0.98]# make
[root@deep Unix-Syslog-0.98]# make all test
[root@deep Unix-Syslog-0.98]# make install
```

Step 10
Convert-TNEF is a Perl module to read TNEF files. TNEF stands for **T**ransport **N**eutral **E**ncapsulation **F**ormat, and if you've ever been unfortunate enough to receive one of these files as an email attachment, you may want to use this module.

```
[root@deep Convert-TNEF-0.17]# perl Makefile.PL
[root@deep Convert-TNEF-0.17]# make
[root@deep Convert-TNEF-0.17]# make all test
[root@deep Convert-TNEF-0.17]# make install
```

Step 11
Convert-UUlib is a versatile and powerful decoder/encoder library for a variety of encodings used in Usenet and Mail (uuencode, xxencode, b64, binhex...). The library contains a variety of heuristics to reliably decode any files found in the input files, whether part of a single mail folder or spread over hundreds of files. Its two-pass architecture makes it possible to decode hundreds of megabytes in one sweep, without using much virtual memory.

```
[root@deep Convert-UUlib-0.212]# perl Makefile.PL
[root@deep Convert-UUlib-0.212]# make
[root@deep Convert-UUlib-0.212]# make all test
[root@deep Convert-UUlib-0.212]# make install
```

Installing all the required compression software's:

Once the Perl modules have been properly installed on your server, it's time to go with the installation of all compression software that we also need for AMaViS to work. You will note that not all the compression software we'll install later is common on regular Linux systems. This is because there is a lot of old compression software used on computers. Though most of these have been replaced with new methods and software.

Now some may say; why we need them if there are old and not used anymore?

Imagine if I use one of this old compression software to send you an email message with a virus in attachment. What will happen? Your Linux mail server will let the message pass the filter because it does not have any idea about the way to uncompress the attachment of the message for checking. Smart people known about this method and try to use it as much as possible when they know that you run an Anti-Virus on your mail server. If you install all of the old compression software on your system with AMaViS, then they will have a nice surprise.

Finally some compression software as described below, simply rely on Windows. We need to install them to be able to deal with file archives compressed using the Windows OS, in this way our server has all the possible compression software it might need to use when it receives a mail attachment.

Installing the ARC software:

ARC stands for file **Ar**chive and **C**ompressor. Long since superseded by default Linux zip/unzip software but useful if you have old "**.**arc" files you need to unpack. This is why we need this old software, because if someone tries to send virus under this file format, AMaViS will be able to handle it.

Step 1
First, we have to get the program (ftp://sunsite.unc.edu/pub/Linux/utils/compress) and copy it to the /var/tmp directory of our Linux system and change to this location before expanding the archive. After that, we have to move into the newly created ARC directory and perform the following steps to compile, optimize and install it.

- This can be done with the following commands:

```
[root@deep /]# cp arc-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf arc-version.tar.gz
[root@deep tmp]# cd arc-5.21/
```

Step 2
Now, we must make a list of files on the system before you install the software and one afterwards then compare them using the **diff** utility to find out what files are placed where and finally install ARC in the system.

```
[root@deep arc-5.21]# cd
[root@deep root]# find /* > ARC1
[root@deep root]# cd /var/tmp/arc-5.21/
[root@deep arc-5.21]# install -m0511 -o root -g root arc /usr/bin/
[root@deep arc-5.21]# install -m0511 -o root -g root marc /usr/bin/
[root@deep arc-5.21]# install -m0440 arc.1.gz /usr/share/man/man1/
[root@deep arc-5.21]# cd
[root@deep root]# find /* > ARC2
[root@deep root]# diff ARC1 ARC2 > ARC-Installed
```

Installing the LHA software:

LHA is a program to compress/expand -LH7- format and AMaViS needs it to be able to handle this kind of file format. Please do not use any LHA version 1.15 (or even 2.x). Use the latest of the 1.14 series. Keep in mind, a broken LHA version will cause problems with exe files, too, as an exe file can be a self-extracting LHA-archive. Therefore, it's really important your LHA is fully functional!

Step 1
First, we have to get the program (http://www2m.biglobe.ne.jp/~dolphin/lha/prog/) and copy it to the /var/tmp directory of our Linux system and change to this location before expanding the archive. After that, we have to move into the newly created LHA directory and perform the following steps to compile, optimize and install it.

- This can be done with the following commands:

```
[root@deep /]# cp lha-1.14i.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf lha-1.14i.tar.gz
[root@deep tmp]# cd lha-1.14i/
```

Step 2
Before going into the compilation of the program, we'll edit the Makefile file to change the default compiler flags to fit our own CPU architecture for better performance and change the default top-level installation directory of the software to reflect our installation location.

- Edit the **Makefile** file (vi +26 Makefile) and change the following lines:

```
OPTIMIZE       = -O2 -DSUPPORT_LH7 -DMKSTEMP
```

To read:

```
OPTIMIZE       = -O2 -march=i686 -funroll-loops -DSUPPORT_LH7 -DMKSTEMP
```

```
BINDIR = /usr/local/bin
MANDIR = /usr/local/man
```

To read:

```
BINDIR = /usr/bin
MANDIR = /usr/share/man
```

Step 3
Now, we must make a list of files on the system before you install the software and one afterwards then compare them using the **diff** utility to find out what files are placed where and finally install LHA.

```
[root@deep lha-1.14i]# make all
[root@deep lha-1.14i]# cd
[root@deep root]# find /* > LHA1
[root@deep root]# cd /var/tmp/lha-1.14i/
[root@deep lha-1.14i]# make install
[root@deep lha-1.14i]# chown 0.0 /usr/bin/lha
[root@deep lha-1.14i]# cd
[root@deep root]# find /* > LHA2
[root@deep root]# diff LHA1 LHA2 > LHA-Installed
```

Installing the TNEF software:

TNEF is a program for unpacking MIME attachments of type "application/ms-tnef". This is a Microsoft only attachment. Due to the proliferation of Microsoft Outlook and Exchange mail servers; more and more mail is encapsulated into this format. The TNEF program allows AMaViS to unpack the attachments, which were encapsulated into the TNEF attachment.

Step 1
First, we have to get the program (http://world.std.com/~damned/software.html) and copy it to the /var/tmp directory of our system and change to this location before expanding the archive. After that, we have to move into the newly created TNEF directory and perform the following steps to compile, configure, optimize and install it.

- This can be done with the following commands:

```
[root@deep /]# cp tnef-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf tnef-version.tar.gz
[root@deep tmp]# cd tnef-1.1/
```

- To configure and optimize TNEF use the following compilation lines:

```
CFLAGS="-O2 -march=i686 -funroll-loops" \
./configure \
--prefix=/usr \
--bindir=/usr/bin \
--mandir=/usr/share/man
```

Step 2
Now, we must make a list of files on the system before you install the software and one afterwards then compare them using the **diff** utility to find out what files are placed where and finally install TNEF in the system.

```
[root@deep tnef-1.1]# make
[root@deep tnef-1.1]# cd
[root@deep root]# find /* > TNEF1
[root@deep root]# cd /var/tmp/tnef-1.1/
[root@deep tnef-1.1]# make install
[root@deep tnef-1.1]# cd
[root@deep root]# find /* > TNEF2
[root@deep root]# diff TNEF1 TNEF2 > TNEF-Installed
```

Installing the UNARJ software:

The UNARJ program is used to uncompress ".arj" format archives. The ".arj" format archive was used mostly on DOS machines. As usually, we install this program to permit AMaViS to deal with this kind of format.

Step 1
First, we have to get the program (ftp://metalab.unc.edu/pub/Linux/utils/compress/) and copy it to the /var/tmp directory of our Linux system and change to this location before expanding the archive. After that, we have to move into the newly created UNARJ directory and perform the following steps to compile, optimize and install it.

- This can be done with the following commands:

```
[root@deep /]# cp unarj-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf unarj-version.tar.gz
[root@deep tmp]# cd unarj-2.43/
```

Step 2
Now, we must make a list of files on the system before you install the software and one afterwards then compare them using the **diff** utility to find out what files are placed where and finally install UNARJ in the system.

```
[root@deep unarj-2.43]# make
[root@deep unarj-2.43]# cd
[root@deep root]# find /* > UNARJ1
[root@deep root]# cd /var/tmp/unarj-2.43/
[root@deep unarj-2.43]# install -m0511 unarj /usr/bin/
[root@deep unarj-2.43]# chown 0.0 /usr/bin/unarj
[root@deep unarj-2.43]# cd
[root@deep root]# find /* > UNARJ2
[root@deep root]# diff UNARJ1 UNARJ2 > UNARJ-Installed
```

Installing the UNRAR software:

The UNRAR utility is used for extracting, testing and viewing the contents of archives created with the RAR archive, version 1.50 and above. Again, AMaViS need this utility to know how to extract archives created with this tool.

Step 1
First, we have to get the program (ftp://sunsite.unc.edu/pub/Linux/utils/compress/) and copy it to the /var/tmp directory of our Linux system and change to this location before expanding the archive. After that, we have to move into the newly created UNRAR directory and perform the following steps to compile, optimize and install it.

- This can be done with the following commands:

```
[root@deep /]# cp unrar-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf unrar-version.tar.gz
[root@deep tmp]# cd unrar-2.71/
```

Step 2
Before going into the compilation of the program, we'll edit some source files to define our operating system, change the default compiler flags to fit our own CPU architecture for better performance and change the top-level installation directory of the software to reflect our installation location.

- Edit the **os.h** file (vi os.h) and change the following lines:

```
#define _UNIX
```

To read:

```
define _UNIX
```

```
#define LITTLE_ENDIAN
```

To read:

```
define LITTLE_ENDIAN
```

- Edit the **Makefile** file (vi +6 Makefile) and change the following lines:

```
BINDIR = /usr/local/bin
```

To read:

```
BINDIR = /usr/bin
```

```
CFLAGS = -D_UNIX -Wall -O2 -fomit-frame-pointer -fno-strength-reduce
```

To read:

```
CFLAGS = -D_UNIX -Wall -O2 -march=i686 -funroll-loops -fno-strength-
reduce
```

Step 3
Now, we must make a list of files on the system before you install the software and one afterwards then compare them using the **diff** utility to find out what files are placed where and finally install UNRAR in the system.

```
[root@deep unrar-2.71]# make
[root@deep unrar-2.71]# cd
[root@deep root]# find /* > UNRAR1
[root@deep root]# cd /var/tmp/unrar-2.71/
[root@deep unrar-2.71]# make install
[root@deep unrar-2.71]# cd
[root@deep root]# find /* > UNRAR2
[root@deep root]# diff UNRAR1 UNRAR2 > UNRAR-Installed
```

Installing the ZOO software:

ZOO is a file archiving utility for maintaining collections of files. It uses Lempel-Ziv compression to provide space savings in the range of 20 to 80 percent depending on the type of data. Unfortunately, this software is not natively available for Linux and we have to patch it to make it work with Linux. The patch file is big and I cannot list it in the book, therefore I've manually patched the software for you and put it on the OpenNA website in ".bz2" and SRPMS format.

Step 1
First, we have to get the program (http://www.openna.com/products/books/securing-optimizing-linux/3rdedition/index.htm) and copy it to the /var/tmp directory of our Linux system and change to this location before expanding the archive. After that, we have to move into the newly created ZOO directory and perform the following steps to compile and install it.

- This can be done with the following commands:

```
[root@deep /]# cp zoo-2.10.tar.bz2 /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xjpf zoo-2.10.tar.bz2
[root@deep tmp]# cd zoo-2.10/
```

Step 2
Now, we must make a list of files on the system before you install the software and one afterwards then compare them using the **diff** utility to find out what files are placed where and finally install ZOO in the system.

```
[root@deep zoo-2.10]# make linux
[root@deep zoo-2.10]# cd
[root@deep root]# find /* > ZOO1
[root@deep root]# cd /var/tmp/zoo-2.10/
[root@deep zoo-2.10]# install -m0511 -s fiz /usr/bin/fiz
[root@deep zoo-2.10]# install -m0511 -s zoo /usr/bin/zoo
[root@deep zoo-2.10]# install -m0440 fiz.1  /usr/share/man/man1
[root@deep zoo-2.10]# install -m0440 zoo.1  /usr/share/man/man1
[root@deep zoo-2.10]# chown 0.0 /usr/bin/fiz
[root@deep zoo-2.10]# chown 0.0 /usr/bin/zoo
[root@deep zoo-2.10]# cd
[root@deep root]# find /* > ZOO2
[root@deep root]# diff ZOO1 ZOO2 > ZOO-Installed
```

Installing the FREEZE software:

FREEZE is another compression program that AMaViS needs when it encounters this kind of file format in mail attachment. As with the above ZOO software, FREEZE has some small problems. The problem with this program is that it's really hard to find on the Internet. I was the lucky to find it when I testing it for AMaViS but don't remember where I found it, therefore I've put a copy of it on the OpenNA website in ".bz2" and SRPMS format.

Step 1
First, we have to get the program (http://www.openna.com/products/books/securing-optimizing-linux/3rdedition/index.htm) and copy it to the `/var/tmp` directory of our Linux system and change to this location before expanding the archive. After that, we have to move into the newly created `FREEZE` directory and perform the following steps to compile, optimize and install it.

- This can be done with the following commands:

```
[root@deep /]# cp freeze-2.5.0.tar.bz2 /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xjpf freeze-2.5.0.tar.bz2
[root@deep tmp]# cd freeze-2.5.0/
```

Step 2
Before going into the configuration and compilation of the program, we'll edit the `Makefile.in` file to change the default compiler flags to fit our own `CPU` architecture for better performance and change the top-level directory for installation of the software to reflect our installation location.

- Edit the **`Makefile.in`** file (`vi Makefile.in`) and change the following lines:

```
CFLAGS        = -I.     # -O2   # for gcc 2.2.2
```

To read:

```
CFLAGS        = -O2 -march=i686 -funroll-loops -I.


prefix        = /usr/local
MANDEST       = $(prefix)/man/man1
```

To read:

```
prefix        = /usr
MANDEST       = $(prefix)/share/man/man1
```

Step 3
Here we simply configure the software for our system.

- To configure `FREEZE` use the following compilation line:

```
./configure -DCOMPAT -DFASTHASH
```

Step 4
Now, we must make a list of files on the system before you install the software and one afterwards then compare them using the **diff** utility to find out what files are placed where and finally install FREEZE in the system.

```
[root@deep freeze-2.5.0]# make
[root@deep freeze-2.5.0]# cd
[root@deep root]# find /* > FREEZE1
[root@deep root]# cd /var/tmp/freeze-2.5.0/
[root@deep freeze-2.5.0]# make install
[root@deep freeze-2.5.0]# chown 0.0 /usr/bin/freeze
[root@deep freeze-2.5.0]# chown 0.0 /usr/bin/melt
[root@deep freeze-2.5.0]# chown 0.0 /usr/bin/unfreeze
[root@deep freeze-2.5.0]# chown 0.0 /usr/bin/fcat
[root@deep freeze-2.5.0]# chown 0.0 /usr/bin/statist
[root@deep freeze-2.5.0]# cd
[root@deep root]# find /* > FREEZE2
[root@deep root]# diff FREEZE1 FREEZE2 > FREEZE-Installed
```

At this stage of our installation, all the required software should now be already present on our mail server where we want to run AMaViS. Now, we can go to the AMaViS installation and configuration part safety.

Compiling - Optimizing & Installing AMaViS

Below are the steps that you must make to configure, and compile the AMaViS software before installing it on your system. First off, we install the program as user "root" so as to avoid any authorization problems.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- This can be done with the following commands:

```
[root@deep /]# cp amavis-perl-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf amavis-perl-version.tar.gz
```

Step 2
In order to check that the version of AMaViS, which you are going to install, is an original and unmodified one, use the command described below to check its MD5 hashes checksum.

- To verify the MD5 checksum of AMaViS, use the following command:

```
[root@deep tmp]# md5sum amavis-perl-11.tar.gz
```

This should yield an output similar to this:

```
e19bfabb2da4aeccc8227766995442d5  amavis-perl-11.tar.gz
```

Now check that this checksum is exactly the same as the one published on the AMaViS website at the following URL: http://www.amavis.org/download.php3

Step 3
AMaViS cannot run as super-user root; for this reason we must create a special user with no shell privileges on the system for running AMaViS daemon.

- To create this special AMaViS user on OpenNA Linux, use the following command:
```
[root@deep tmp]# groupadd -g 45 amavis > /dev/null 2>&1 || :
[root@deep tmp]# useradd -c "Virus Scanning Interface" -d
/var/lib/amavis -g 45 -s /bin/false -u 45 amavis > /dev/null 2>&1 || :
```

- To create this special AMaViS user on Red Hat Linux, use the following command:
```
root@deep tmp]# groupadd -g 45 amavis > /dev/null 2>&1 || :
[root@deep tmp]# useradd -u 45 -g 45 -s /bin/false -M -r -d
/var/lib/amavis amavis > /dev/null 2>&1 || :
```

The above command will create a null account, with no password, no valid shell, no files owned-nothing but a UID and a GID for the program. Remember that AMaViS daemon does not need to have a shell account on the server.

Step 4
Now, edit the **shells** file (vi /etc/shells) and add a non-existent shell name "/bin/false", which is the one we used in the useradd command above.

```
[root@deep tmp]# vi /etc/shells
/bin/bash2
/bin/bash
/bin/sh
/bin/false ← This is our added no-existent shell
```

Step 5
Next, move into the newly created AMaViS directory and perform the following steps to configure and compile the software for your system.

- To move into the newly created AMaViS directory use the following command:
```
[root@deep tmp]# cd amavis-perl-11/
```

Step 6
There is a bug in Archive-Tar Perl program that we have installed previously to work with AMaViS. Here is a work around bug to fix the Archive-Tar problem.

- Edit the **amavis.in** file (vi +583 amavis/amavis.in) and change the lines:

```
my $tar = Archive::Tar->new("$TEMPDIR/parts/$part") ||
        do_exit($REGERR, __LINE__);
```

To read:

```
my $tar = eval { Archive::Tar->new("$TEMPDIR/parts/$part") };

unless (defined($tar)) {
        do_log(4,"Faulty archive $part");
        return 0;
}
```

Step 7
Once the modifications have been made to the AMaViS source file as shown above, it is time to configure and compile AMaViS for our system.

- To configure and optimize AMaViS for your system use the following compilation lines:

```
CFLAGS="-O2 -march=i686 -funroll-loops"; export CFLAGS
./configure \
--prefix=/usr \
--sbindir=/usr/sbin \
--sysconfdir=/etc \
--localstatedir=/var \
--mandir=/usr/share/man \
--with-mailto=postmaster \
--with-amavisuser=amavis \
--with-sophos-ide=/usr/lib/sophos \
--with-runtime-dir=/var/lib/amavis \
--with-virusdir=/var/lib/amavis/virusmails \
--with-warnrecip \
--enable-syslog
```

This tells AMaViS to set itself up for this particular configuration setup with:

- Inform AMaViS to send emails to postmaster on the system.
- Run AMaViS as the user "amavis" who have already created.
- Inform AMaViS where the Sophos IDE files are installed on the system.
- Where the directory for runtime files is located.
- Where the quarantine directory for infected mail is located.
- Send notifications to receiver(s) when a virus is detected.
- Use syslog to log system messages.

Step 8
At this stage the program is ready to be built and installed. We build AMaViS with the 'make' command and produce a list of files on the system before we install the software, and one afterwards, then compare them using the **diff** utility to find out what files were placed where and finally install AMaViS.

```
[root@deep amavis-perl-11]# make
[root@deep amavis-perl-11]# cd
[root@deep root]# find /* > AMaViS1
[root@deep root]# cd /var/tmp/amavis-perl-11/
[root@deep amavis-perl-11]# make install
[root@deep amavis-perl-11]# cd
[root@deep root]# find /* > AMaViS2
[root@deep root]# diff AMaViS1 AMaViS2 > AMaViS-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

Step 9
Once the compilation and installation of AMaViS have been accomplished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete AMaViS and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf amavis-perl-11/
[root@deep tmp]# rm -f amavis-perl-version.tar.gz
```

Running AMaViS with Exim

This section applies only if you want to run AMaViS with Exim. Like SpamAssassin, AMaViS after its default installation cannot automatically scan incoming or outgoing mail messages. You have to do it manually, and this is not what we want to do, therefore we have to integrate it with Exim. In this way, Exim will automatically call AMaViS to scan for possible viruses every time it receives or sends mail messages on our server.

Necessary steps to integrate AMaViS with Exim:

Procedures to allow AMaViS to run with Exim is not difficult to accomplish since everything will happen inside the exim.conf file but we have to be careful of the order in which the additional configuration lines related to AMaViS should be added to the configuration file. This is very important also don't forget to add the "amavis" user to the file.

Step 1
First, we have to edit our default exim.conf file and look for the line that's reads (trusted_users = mail). We have to change it to include the "amavis" UID.

- Edit **exim.conf** file (vi /etc/mail/exim.conf) and change the following line.

```
trusted_users = mail
```

```
To read:
```

```
trusted_users = mail:amavis
```

Step 2
Next, we have to include new router conditions related to AMaViS. This is done by adding the following lines at the top of the "Routers Configuration Section" of exim.conf file. Text in bold is what we have added to the default configuration file.

Add the following lines at the TOP of the "**Routers Configuration Section**".

- Edit **exim.conf** file (vi /etc/mail/exim.conf) and add the following lines at the top of the "Routers Configuration Section" as follow.

```
begin routers
```

```
amavis_router:
  driver = accept
  condition = "${if or{ {eq {$received_protocol}{scanned-ok}} \
  {eq {$received_protocol}{spam-scanned}} } {0}{1}}"
  retry_use_local_part
  transport = amavis
```

```
spamcheck_router:
  no_verify
  check_local_user
  condition = "${if and { {!def:h_X-Spam-Flag:} \
  {!eq {$received_protocol}{spam-scanned}}} {1}{0}}"
  driver = accept
  transport = spamcheck
```

NOTE: As you can see, we assume that `SpamAssassin` is already included in the configuration file. This is very important, because our `AMaViS` parameters take in consideration that `SpamAssassin` parameters are included into the configuration file of `Exim`. In this way, `AMaViS` will first scan the message for possible viruses and then pass the mail to `SpamAssassin` to check for possible Spam before delivering the message to its final recipient.

Step 3
Now, we have to include the transports parameters for `AMaViS`. This is done by adding the following lines to the end of the “Transports Configuration Section” of the `exim.conf` file. Text in bold is what we have added to the default configuration file.

Add the following lines at the END of the “**Transports Configuration Section**”.

- Edit **`exim.conf`** file (`vi /etc/mail/exim.conf`) and add the following lines at the end of the "Transports Configuration Section" as follow.

```
spamcheck:
  driver = pipe
  batch_max = 100
  command = /usr/sbin/exim -oMr spam-scanned -bS
  use_bsmtp = true
  transport_filter = /usr/bin/spamc
  home_directory = "/tmp"
  current_directory = "/tmp"
  user = mail
  group = mail
  log_output = true
  return_fail_output = true
  return_path_add = false
  message_prefix =
  message_suffix =

amavis:
  driver = pipe
  check_string =
  command = "/usr/sbin/amavis -f <${sender_address}> -d
${pipe_addresses}"
  current_directory = "/var/lib/amavis"
  escape_string =
  group = amavis
  headers_add = "X-Virus-Scanned: by AMaViS OpenNA Linux"
  message_prefix =
  message_suffix =
  path = "/bin:/sbin:/usr/bin:/usr/sbin"
  no_return_output
  no_return_path_add
  user = amavis
```

Step 4
Now, we have to restart the Exim daemon for the changes to take effect.

- To restart Exim, use the following command:
```
[root@deep /]# /etc/init.d/exim restart
Shutting down Exim:              [OK]
Starting Exim:                   [OK]
```

Running AMaViS with Qmail

This section applies only if you want to run AMaViS with Qmail. AMaViS after its default installation cannot automatically filter incoming or outgoing mail messages. You have to do it manually, and this is not what we want to do, therefore we have to automate the procedure. Integrating AMaViS with Qmail can do this. In this way, our Qmail server will automatically call AMaViS every time it sends or receives mail messages for users on our mail server.

Necessary steps to integrate AMaViS with Qmail:

As you might imagine, the integration of Qmail with AMaViS is completely different from the way we did it with Exim and you can also be sure that you will have to carry out some special steps to achieve the results, due to the very non-standard way Qmail communicates with AMaViS. But don't worry, the steps are really not difficult to accomplish and once finished, it's really impressive to see your own Qmail server easily deal with the Anti-Virus interface for all mail users on the server.

Step 1
When Qmail is running, it uses its qmail-queue program to queue a mail message for delivery. When we want to make AMaViS work with Qmail, we have to change the way it delivers mail messages. To do this, we should rename our existing qmail-queue program to become qmail-queue-real and move the AMaViS script to qmail-queue. In this way, every message on the system, either sent or received, will be scanned by AMaViS before being delivered to its final recipient. Once the AMaViS script is moved to become qmail-queue, we should make sure it has the same permissions as qmail-queue-real for Qmail to be able to run it.

- This can be done with the following commands:
```
[root@deep /]# cd /usr/bin/
[root@deep bin]# mv qmail-queue qmail-queue-real
[root@deep bin]# mv /usr/sbin/amavis /usr/bin/qmail-queue
[root@deep bin]# chown qmailq.qmail qmail-queue
[root@deep bin]# chmod 4511 qmail-queue
```

Step 2
Next, we have to change the ownership of both AMaViS quarantine directories to be owned by the "qmailq" UID.

- This can be done with the following commands:
```
[root@deep /]# cd /var/lib/
[root@deep lib]# chown -R qmailq.root amavis
```

Step 3
On most Linux systems the binary program called "suidperl" is not SUID for security reasons, with Qmail and AMaViS, it should be, otherwise the Anti-Virus interface won't work.

- To re-enable it use the following commands:
```
[root@deep /]# chown root.qmail /usr/bin/suidperl
[root@deep /]# chmod 4511 /usr/bin/suidperl
```

Step 4
Finally, we should create a new alias file for user "amavis" under the /etc/mail/alias directory on our mail server.

- This can be done with the following commands:
```
[root@deep /]# touch /etc/mail/alias/.qmail-amavis
[root@deep /]# chmod 644 /etc/mail/alias/.qmail-amavis
[root@deep /]# chown root.nofiles /etc/mail/alias/.qmail-amavis
```

Testing AMaViS

Once our Anti-Virus interface software is installed, we have to run a test to make sure AMaViS is working correctly on our system. The test should complete successfully or you will eventually have problems scanning email messages for possible virus infections. The test consists of sending a virus to our mail server and see if AMaViS detects it.

To be able to successfully make the test, we have to move inside the AMaViS source subdirectory called "tests" and run the "make EICAR.COM" command that will created a file called "EICAR.COM" with the virus checker test pattern. This virus file will be used to test AmaViS's functionality. You don't have to modify the contents of this text file, just use it.

- To move into the AMaViS source subdirectory "tests" use the following command:
```
[root@deep /]# cd /var/tmp/amavis-perl-11/tests/
```

- To create the "EICAR.COM" virus, use the following command:
```
[root@deep tests]# make EICAR.COM
```

Test 1 – Sending a virus to our mail server
In this test, we will send the "EICAR.COM" virus we have created previously to the "root" user account on our mail server to verify if AMaViS can correctly detect it and put it into quarantine.

- To send the virus, use the following command:
```
[root@deep tests]# sendmail root < EICAR.COM
```

If everything is ok, you will now have three messages waiting for you in your mailbox. If you encounter problems, look at the log file (/var/log/maillog) to see if there is any relevant information there. The above test should work for both Exim and Qmail without any problems.



Part XII

Database Server

CHAPTER 38

MySQL

IN THIS CHAPTER

1. Compiling - Optimizing & Installing `MySQL`
2. Configuring `MySQL`
3. Securing `MySQL`
4. Optimizing `MySQL`
5. `MySQL` Administrative Tools

Linux MySQL

Abstract

Once you decide to go into serious business, you'll inevitably find that you need a database to store/retrieve information. One of the primary reasons for the invention of computer was to store, retrieve and process information and to do all this very quickly. The most popular database systems are based on the **I**nternational **S**tandard **O**rganization (ISO) SQL specifications which are also based on ANSI SQL (American) standards.

This part of the book will deal with software other than the one's which the Linux distribution, may or may not provide as a part of its core distribution. In some cases it may be provided as an extra but may also come as a pre-compiled binary, which may not exactly suit your purpose. Hence we have, in most cases, used source packages, usually packed as tar gzipped -*.tar.gz. This gives us the maximum amount of choice to tweak, secure, optimize and delete the options within this software.

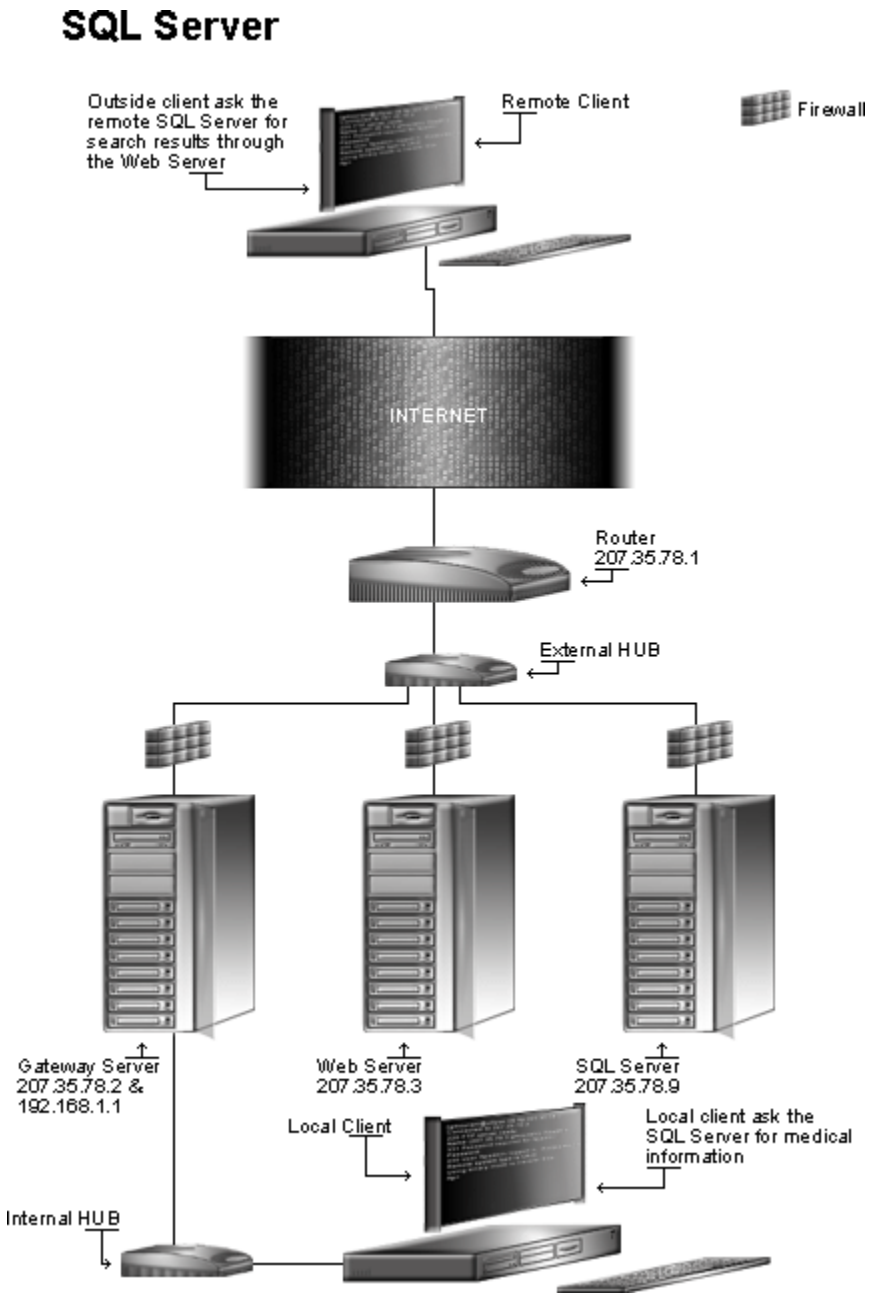
Once you begin to serve, and supply services to your customers, you'll inevitably find that you need to keep information about them in an archive, which has to be accessible and able to be modified at any time. These tasks can be accomplished with the use of a database.

There are many databases are available for Linux; choosing one can be complicated, as it must be able to support a number of programming languages, standards and features. PostgreSQL is a sophisticated Object-Relational DBMS and supports almost all SQL constructs, which may respond to complicated and complex database needs.

In real use, and especially for Web server connectivity with SQL databases, the need for this kind of complex arrangement is not always true and may penalize performance. For these reasons some companies decide to create an SQL server which responds to these requirements. MySQL is a small SQL database built with the most essential SQL constructs only and increases performance by eliminating functions.

MySQL is a true multi-user, multi-threaded SQL database server. SQL (**S**tructured **Q**uery **L**anguage) is the most popular and standardized database language in the world. MySQL is a client/server implementation that consists of a server daemon "mysqld" and many different client programs and libraries.

The main goals of MySQL are speed, robustness and ease of use. MySQL was originally developed due to the need of an SQL server that could handle very large databases an order of magnitude faster than what any database vendor could offer on inexpensive hardware.



This schema shows you some possible uses of `SQL` servers.

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, at personal discretion).
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest MySQL version number is 3.23.51

The procedures given in this chapter are likely to work on all Linux platforms, but we have only tested it on OpenNA Linux and Red Hat Linux.

Packages

The following is based on information listed by MySQL as of 2002/06/24. Please regularly check http://www.mysql.org/ for the latest status. We chose to install the required component from a source file because it provides the facility to fine tune the installation.

Source code is available from:
MySQL Homepage: http://www.mysql.org/
MySQL FTP Site: 64.28.67.70
You must be sure to download: mysql-3.23.51.tar.gz

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all the files installed on the system in the eventuality of an update in the future. To solve the problem, it is a good idea to make a list of files on the system before you install MySQL, and then one afterwards, and then compare them using the **diff** utility to find out what files were placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep root]# find /* > MySQL1
  ```
- And the following one after you install the software:
  ```
  [root@deep root]# find /* > MySQL2
  ```
- Then use the following command to get a list of what changed:
  ```
  [root@deep root]# diff MySQL1 MySQL2 > MySQL-Installed
  ```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. In the example above, we use the /root directory of the system to store all generated list files.

Compiling - Optimizing & Installing MySQL

Below are the steps that you must make to configure, compile and optimize the MySQL software before installing it onto your system. First off, we install the program as the user "root" so as to avoid permissioning problems.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- This can be done with the following commands:

```
[root@deep /]# cp mysql-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf mysql-version.tar.gz
```

Step 2
We must create a new user account called "mysql" with shell privileges on the system to be the owner of the MySQL database files and daemon.

- To create this special MySQL user on OpenNA Linux, use the following command:

```
[root@deep tmp]# groupadd -g 27 mysql > /dev/null 2>&1 || :
[root@deep tmp]# useradd -c "MySQL Server" -d /var/lib/mysql -g 27 -m -s
/bin/bash -u 27 mysql > /dev/null 2>&1 || :
```

- To create this special MySQL user on Red Hat Linux, use the following command:

```
[root@deep tmp]# groupadd -g 27 mysql > /dev/null 2>&1 || :
[root@deep tmp]# useradd -u 27 -g 27 -s /bin/bash -M -r -d
/var/lib/mysql mysql > /dev/null 2>&1 || :
```

The above command will create a real account, with no password, but valid shell access for the user mysql because we need it to connect to the database on the terminal of the server.

Step 3
Next, move into the newly created MySQL source directory and perform the following steps to configure and optimize MySQL for your system.

- To move into the newly created MySQL source directory use the following command:

```
[root@deep tmp]# cd mysql-3.23.51/
```

- To configure and optimize MySQL use the following compilation lines:

```
CFLAGS="-static -O2 -march=i686 -funroll-loops" \
CXXFLAGS="-static -O2 -march=i686 -funroll-loops -felide-constructors -
fno-exceptions -fno-rtti" \
./configure \
--prefix=/usr \
--libexecdir=/usr/sbin \
--sysconfdir=/etc \
--localstatedir=/var/lib/mysql \
--mandir=/usr/share/man \
--disable-shared \
--enable-assembler \
--with-thread-safe-client \
--with-mysqld-user="mysql" \
--with-unix-socket-path=/var/lib/mysql/mysql.sock \
--with-client-ldflags=-all-static \
--with-mysqld-ldflags=-all-static \
--without-readline \
--without-debug \
--without-docs \
--without-bench
```

This tells MySQL to set itself up for this particular configuration setup with:

- Disable the build of shared libraries for improved performance of the software.
- Use assembler versions of some string functions.
- Compile the client part of the software with threads support, again for better performance.
- Define the user under which we should run the database as.
- Use Unix sockets rather than TCP/IP to connect to a database for better performance.
- Use system readline instead of bundled copy.
- Build a production version without debugging code to run MySQL 20% faster for most queries.
- Skip building of the MySQL help documentations to save space on the server.
- Skip building of the benchmark tools to save space on the server.

Step 4
Now, we must make a list of files on the system before installing the software and one afterwards then compare them using the **diff** utility to find out what files are placed where and finally install MySQL.

```
[root@deep mysql-3.23.51]# make
[root@deep mysql-3.23.51]# cd
[root@deep root]# find /* > MySQL1
[root@deep root]# cd /var/tmp/mysql-3.23.51/
[root@deep mysql-3.23.51]# make install
[root@deep mysql-3.23.51]# mkdir -p /var/run/mysqld
[root@deep mysql-3.23.51]# chown mysql.mysql /var/run/mysqld
[root@deep mysql-3.23.51]# rm -rf /usr/mysql-test/
[root@deep mysql-3.23.51]# rm -f /usr/share/mysql/mysql-*.spec
[root@deep mysql-3.23.51]# rm -f /usr/share/mysql/mysql-log-rotate
[root@deep mysql-3.23.51]# strip /usr/sbin/mysqld
[root@deep mysql-3.23.51]# cd
[root@deep root]# find /* > MySQL2
[root@deep root]# diff MySQL1 MySQL2 > MySQL-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

Step 5
At this stage, all the files and binaries related to MySQL database have been installed onto your computer. It is time to verify if the mysqld daemon is linked statically as we want it to be.

- To verify if the mysqld daemon is linked statically, use the following command:
```
[root@deep /]# ldd /usr/sbin/mysqld
        not a dynamic executable
```

If the result of the command is the same as the one shown above, then congratulations! All libraries required by the daemon to successfully run on your server have been compiled directly into the mysqld binary.

Step 6
Once the compilation, optimization and installation of the software has completed, we can free up some disk space by deleting the program tar archive and the related source directory, since they are no longer needed.

- To delete MySQL and its related source directory, use the following commands:
```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf mysql-version/
[root@deep tmp]# rm -f mysql-version.tar.gz
```

The rm command as used above will remove all the source files we have used to compile and install MySQL. It will also remove the MySQL compressed archive from the /var/tmp directory.

Configuring MySQL
After MySQL has been built and installed successfully on your system, the next step is to configure and customize its configuration files to fit your needs.

- ✓ /etc/my.cnf: (The MySQL Configuration File)
- ✓ /etc/logrotate.d/mysqld: (The MySQL Log Rotation File)
- ✓ /etc/init.d/mysqld: (The MySQL Initialization File)

/etc/my.cnf: The MySQL Configuration File
The /etc/my.cnf file is the main configuration file for MySQL. It is in this configuration file that MySQL gets all of its information, such as the directory where databases are stored, where mysqld socket live and the user under which the mysqld daemon will run, etc.

Step 1
This file is checked to get the required information each time the database starts its daemon. It is also used to specify optimization parameters for the database, but for the moment you can add the lines shown below, and later in this chapter we give more information about other possible parameters, particularly the ones related to optimization, that we could add to this file.

- Create the **my.cnf** file (`touch /etc/my.cnf`) and add the following lines:

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock

[mysql.server]
user=mysql
basedir=/var/lib

[safe_mysqld]
err-log=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
```

Step2
Now, set the permissions of the `my.cnf` file to be (`0644/-rw-r--r--`) and owned by the super-user '`root`' for security reasons.

- To change the permissions and ownership of the `my.cnf` file, use:

```
[root@deep /]# chmod 644 /etc/my.cnf
[root@deep /]# chown 0.0 /etc/my.cnf
```

/etc/logrotate.d/mysqld: The MySQL Log Rotation File

The `/etc/logrotate.d/mysqld` file allows the `MySQL` database server to automatically rotate its log files at the specified time.

Step1
Here we'll configure the `/etc/logrotate.d/mysqld` file to rotate each week its log files automatically.

- Create the **mysqld** file (`touch /etc/logrotate.d/mysqld`) and add the lines:

```
/var/log/mysqld.log {
    missingok
    create 0640 mysql mysql
    prerotate
        [ -e /var/lock/subsys/mysqld ] && /bin/kill -HUP `/bin/cat
/var/run/mysqld/mysqld.pid` || /bin/true
    endscript
    postrotate
        [ -e /var/lock/subsys/mysqld ] && /bin/kill -HUP `/bin/cat
/var/run/mysqld/mysqld.pid` || /bin/true
    endscript
}
```

Step2
Now, set the permissions of the `mysqld` file to be (`0644/-rw-r--r--`) and owned by the super-user '`root`' for security reasons.

- To change the permissions and ownership of the `mysqld` file, use the commands:
```
[root@deep /]# chmod 640 /etc/logrotate.d/mysqld
[root@deep /]# chown 0.0 /etc/logrotate.d/mysqld
```

`/etc/init.d/mysqld`: The `MySQL` Initialization File

The `/etc/init.d/mysqld` script file is responsible for automatically starting and stopping the `MySQL` server. Loading the `mysqld` daemon as a standalone daemon will eliminate load time and will even reduce swapping since non-library code will be shared.

Please note that the following script is only suitable for Linux operating systems using `System V`. If your Linux system uses some other method, like `BSD`, you'll have to adjust the script below to make it work for you.

Step 1
Create the **`mysqld`** script file (`touch /etc/init.d/mysqld`) and add the following lines:

```
#!/bin/bash

# This shell script takes care of starting and stopping MySQL.
#
# chkconfig: 345 78 12
# description:  MySQL is a fast & secure SQL database server.
#
# processname: mysqld
# config: /etc/my.cnf
# pidfile: /var/run/mysqld/mysqld.pid

# Source function library.
. /etc/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Source for additional options if we have them.
if [ -f /etc/sysconfig/mysql ] ; then
        . /etc/sysconfig/mysql
fi

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

# If MySQL is not available stop now.
[ -f /usr/bin/safe_mysqld ] || exit 0

# Path to the MySQL binary.
safe_mysqld=/usr/bin/safe_mysqld

RETVAL=0
prog="MySQL"

start() {
        if [ ! -d /var/lib/mysql/mysql ] ; then
            action $"Initializing $prog database: " /usr/bin/mysql_install_db
```

```
            ret=$?
            chown -R mysql.mysql /var/lib/mysql
                if [ $ret -ne 0 ] ; then
                        return $ret
                fi
        fi

        chown -R mysql.mysql /var/lib/mysql
        chmod 0755 /var/lib/mysql
        daemon $safe_mysqld --defaults-file=/etc/my.cnf >/dev/null 2>&1 &
        ret=$?

        if [ $ret -eq 0 ]; then
            action $"Starting $prog: " /bin/true
        else
            action $"Starting $prog: " /bin/false
        fi
        [ $ret -eq 0 ] && touch /var/lock/subsys/mysqld
        return $ret
}

stop() {
        kill `cat /var/run/mysqld/mysqld.pid  2> /dev/null ` > /dev/null 2>&1
        ret=$?

        if [ $ret -eq 0 ]; then
            action $"Shutting down $prog: " /bin/true
        else
            action $"Shutting down $prog: " /bin/false
        fi

        [ $ret -eq 0 ] && rm -f /var/lock/subsys/mysqld
        [ $ret -eq 0 ] && rm -f /var/lib/mysql/mysql.sock
        return $ret
}

# See how we were called.
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  status)
        status mysqld
        RETVAL=$?
        ;;
  reload)
        [ -e /var/lock/subsys/mysqld ] && mysqladmin reload
        RETVAL=$?
        ;;
  restart)
        stop
        start
        RETVAL=$?
        ;;
  condrestart)
        if [ -f /var/lock/subsys/mysqld ]; then
                stop
                start
                RETVAL=$?
        fi
```

```
        ;;
  *)
        echo $"Usage: $0 {start|stop|status|reload|restart|condrestart}"
        exit 1
esac
exit $?
```

Step 2
Once the mysqld script file has been created, it is important to make it executable, change its default permissions, create the necessary links and start it. Making this file executable will allow the system to run it, changing its default permission is to allow only the root user to change this file for security reasons, and creation of the symbolic links will let the process control initialization start the program automatically for you at each system boot.

- To make this script executable and to change its default permissions, use the commands:
```
[root@deep /]# chmod 700 /etc/init.d/mysqld
[root@deep /]# chown 0.0 /etc/init.d/mysqld
```

- To create the symbolic rc.d links for MySQL, use the following commands:
```
[root@deep /]# chkconfig --add mysqld
[root@deep /]# chkconfig --level 345 mysqld on
```

- To start MySQL software manually, use the following command:
```
[root@deep /]# /etc/init.d/mysqld start
Starting MySQL:               [OK]
```

Step 3
Once the SQL server has been started, it's time to assign a password to the super-user of this database. With MySQL server, this user is called, by default 'root', but be aware that MySQL 'root' user has nothing in common with the Unix 'root' user, only the name are the same and NOTHING else.

For security reasons, it's important to assign a password to the MySQL root user, since by default after the installation of the SQL server, the initial root password is empty and allows anyone to connect with this name and therefore do anything to the database.

- To specify a password for the MySQL root user, perform the following actions.
```
[root@deep /]# mysql -u root mysql
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1 to server version: 3.23.49

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> SET PASSWORD FOR root=PASSWORD('mypasswd');
Query OK, 0 rows affected (0.00 sec)

mysql> \q
Bye
```

The value **'mypasswd'** as shown above is where you put the password you want to assign to the MySQL root user (this is the only value you must change in the above command). Once the root password has been set you must, in the future, supply this password to be able to connect as root to the SQL database.

Securing MySQL

This section deals specificlly with actions we can take to improve and tighten security under the MySQL database. The interesting point here is that we refer to the features available within the base installed program and not to any additional software.

Protect the MySQL communication socket:

The unix-domain socket "mysql.sock" which is used to connect to the MySQL database has, by default, the following permissions (0777/srwxrwxrwx), this means that anyone can delete this socket and if this happens, then no-one will be able to connect to your database.

To avoid deletion of the MySQL communication socket under /var/lib/mysql/mysql.sock, you can protect its /var/lib/mysql directory by setting the sticky bit on it.

- To protect and set the sticky bit on directory where the file reside, use the command:
```
[root@deep /]# chmod +t /var/lib/mysql
```

This command will protect your /var/lib/mysql directory so that files can be deleted only by their owners or the super-user (root).

- To check if the sticky bit is set on this directory, use the following command:
```
[root@deep /]# ls -ld /var/lib/mysql
drwxr-xr-t    4 mysql    mysql        1024 May 29 15:00 /var/lib/mysql
```

If the last permission bit is "t", then the bit is set. Congratulations!

Delete the anonymous database:

When you install MySQL server, the program creates two databases by default. The first database is called "mysql" and it's used to hold all the settings of the MySQL server, users, passwords, privileges etc. The second database called "test" is used for testing your SQL database. Any local user can connect, without a password, to this database and do anything.

This database is not needed by the MySQL server to work and can be removed safety.

- To remove the "test" database from your SQL server, use the following command:
```
[root@deep /]$ mysqladmin drop test -p
Enter password:
Dropping the database is potentially a very bad thing to do.
Any data stored in the database will be destroyed.

Do you really want to drop the 'test' database [y/N] y
Database "test" dropped


[root@deep /]# mysql -u root mysql -p
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3 to server version: 3.23.49

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

```
mysql> DELETE FROM db WHERE Db = "test";
Query OK, 1 row affected (0.00 sec)

mysql> DELETE FROM db WHERE Db = "test\_%";
Query OK, 1 row affected (0.00 sec)

mysql> \q
Bye
```

Optimizing MySQL

This section deals specifically with actions we can make to improve and tighten performance of MySQL database. Note that we refer to the features available within the base installed program.

Get some fast SCSI hard disk:

One of the most important parts of optimizing MySQL server, as well as for the majority of all SQL databases, is the speed of your hard disk, the faster it is, and the faster your databases will run. Consider using a SCSI disk with low seek times, like 4.2ms, which can make all the difference, much better performance can also be had with RAID technology.

Skip the updating of the last access time:

As you should know by now, the **noatime** attribute of Linux eliminates the need by the system to make writes to the file system for files. Mounting the file system where your MySQL databases live with the noatime attribute will avoid some disk seeks and will improve the performance of you SQL server.

If you want to mount the file system of the MySQL database with the noatime attribute, it's important to create and install the MySQL database in this partition. In our example, we have created this partition earlier in chapter 2 " and this partition is located on /var/lib.

Step 1
To mount the file system of the MySQL databases with the noatime option, you must edit the **fstab** file (vi /etc/fstab) and add to the line that refers to the /var/lib file system the noatime option after the defaults option as shown below:

- Edit the **fstab** file (vi /etc/fstab), and change the line:

  ```
  LABEL=/var/lib    /var/lib        ext3    defaults        1 2
  ```

 To read:

  ```
  LABEL=/var/lib    /var/lib        ext3    defaults,noatime        1 2
  ```

NOTE: The line relating to /var/lib in your /etc/fstab file could be different from the one above, as this is just an example.

Step 2
Once you have made the necessary adjustments to the `/etc/fstab` file, it is time to inform the Linux system about the modifications.

- This can be accomplished with the following commands:
```
[root@deep /]# mount /var/lib -oremount
```

Each file system that has been modified must be remounted with the command as shown above.

Step 3
After your file system has been remounted, it is important to verify if the modification in the `fstab` file have been correctly applied to the system.

- You can verify if the modification has been correctly applied with the following command:
```
[root@deep /]# cat /proc/mounts
/dev/root     /           ext3    rw 0 0
/proc         /proc       proc    rw 0 0
/dev/sda1     /boot       ext3    rw 0 0
/dev/sda9     /chroot     ext3    rw 0 0
/dev/sda8     /home       ext3    rw 0 0
/dev/sda13    /tmp        ext3    rw 0 0
/dev/sda7     /usr        ext3    rw 0 0
/dev/sda11    /var        ext3    rw 0 0
/dev/sda12    /var/lib    ext2    rw,noatime 0 0
none          /dev/pts    devpts  rw 0 0
```

This command will show you all file systems in your Linux server with parameters applied to them. If you see something like:

```
/dev/sda12 /var/lib   ext3    rw,noatime 0 0
```
Congratulations!

> **NOTE:** Look at the chapter related to the Linux Kernel for more information about the `noatime` attribute and other tunable parameters.

Give MySQL more memory to get better performance:

There are four options and configuration variables directly related to the speed of the `MySQL` database that you might want to tune during server startup. The **`key_buffer_size`** parameter is one of the most important tunable variables; it represents the size of the buffer used for the index blocks by `MySQL` server.

The second is **`table_cache`**, which represents the number of open tables for all threads. By increasing this value, you'll increase the number of file descriptors that `mysqld` requires. The two last variables are **`sort_buffer`**, which speeds up the ORDER BY or GROUP BY operations of the database and **`record_buffer`**, which improves the speed when you do many sequential scans.

Step 1
Depending of the amount of memory, RAM, you have in your system and according to the MYSQL recommendations:

If you have a large amount of memory (>=256M), many tables and want maximum performance with a moderate number of clients, you should use something like this in your my.cnf file:

```
set-variable    = key_buffer=64M
set-variable    = table_cache=256
set-variable    = sort_buffer=4M
set-variable    = record_buffer=1M
```

If you have only 128M and only a few tables, but you still do a lot of sorting, you can use something like this in your my.cnf file:

```
set-variable    = key_buffer=16M
set-variable    = sort_buffer=1M
```

If you have little memory and lots of connections use something like this in your my.cnf file:

```
set-variable    = key_buffer=512k
set-variable    = sort_buffer=100k
set-variable    = record_buffer=100k
```

or even:

```
set-variable    = key_buffer=512k
set-variable    = sort_buffer=16k
set-variable    = table_cache=32
set-variable    = record_buffer=8k
set-variable    = net_buffer=1K
```

These are just some examples, a complete list of tunable parameters depending on your type of SQL server exist under the /usr/share/mysql directory and are available for you to learn. In total there are four example files with lots of tunable parameters for huge, large, medium, and small systems and there are called respectively: my-huge.cnf, my-large.cnf, my-medium.cnf, my-small.cnf. Please, check them to see if one of them better fits your optimization requirements.

Step2
Once you know the values you need for your MySQL database server, it's time to set them in your /etc/my.cnf file. Recall that this file is read each time your database server starts.

In our example below, we will configure the /etc/my.cnf file for a medium system with little memory (32M - 64M) where MySQL plays a important part and systems up to 128M where MySQL is used together with other programs (like a web server). The text in bold is the parts of the configuration file that must be customized and adjusted to satisfy our needs.

- Edit your **my.cnf** file (vi /etc/my.cnf) and enter the values that you have chosen.

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
skip-locking
set-variable    = key_buffer=16M
set-variable    = max_allowed_packet=1M
set-variable    = table_cache=64
set-variable    = sort_buffer=512K
set-variable    = net_buffer_length=8K
set-variable    = myisam_sort_buffer_size=8M

[mysql.server]
user=mysql
basedir=/var/lib

[safe_mysqld]
err-log=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid

[isamchk]
set-variable    = key_buffer=20M
set-variable    = sort_buffer=20M
set-variable    = read_buffer=2M
set-variable    = write_buffer=2M

[myisamchk]
set-variable    = key_buffer=20M
set-variable    = sort_buffer=20M
set-variable    = read_buffer=2M
set-variable    = write_buffer=2M
```

Step 3

- Restart the MySQL database server for the changes to take effect:

```
[root@deep /]# /etc/init.d/mysqld restart
Enter password:
Stopping MySQL:               [OK]
Starting MySQL:               [OK]
```

Step 4
Now you should verify your new values with the mysqladmin command as shown below. One function of this command allows you to see what values a running MySQL server is using.

- To verify the new variables entered in your startup file, use the following command:

```
[root@deep /]# mysqladmin variables -p
Enter password:
```

```
+--------------------------+--------------------------------------------+
| Variable_name            | Value                                      |
+--------------------------+--------------------------------------------+
| ansi_mode                | OFF                                        |
| back_log                 | 50                                         |
| basedir                  | /usr/                                      |
| binlog_cache_size        | 32768                                      |
| character_set            | latin1                                     |
| character_sets           | latin1 dec8 dos german1 hp8 koi8_ru latin2 |
| concurrent_insert        | ON                                         |
| connect_timeout          | 5                                          |
| datadir                  | /var/lib/mysql/                            |
| delay_key_write          | ON                                         |
| delayed_insert_limit     | 100                                        |
| delayed_insert_timeout   | 300                                        |
| delayed_queue_size       | 1000                                       |
| flush                    | OFF                                        |
| flush_time               | 0                                          |
| have_bdb                 | NO                                         |
| have_gemini              | NO                                         |
| have_innobase            | NO                                         |
| have_isam                | YES                                        |
| have_raid                | NO                                         |
| have_ssl                 | NO                                         |
| init_file                |                                            |
| interactive_timeout      | 28800                                      |
| join_buffer_size         | 131072                                     |
| key_buffer_size          | 16773120                                   |
| language                 | /usr/share/mysql/english/                  |
| large_files_support      | ON                                         |
| locked_in_memory         | OFF                                        |
| log                      | OFF                                        |
| log_update               | OFF                                        |
| log_bin                  | OFF                                        |
| log_slave_updates        | OFF                                        |
| long_query_time          | 10                                         |
| low_priority_updates     | OFF                                        |
| lower_case_table_names   | 0                                          |
| max_allowed_packet       | 1047552                                    |
| max_binlog_cache_size    | 4294967295                                 |
| max_binlog_size          | 1073741824                                 |
| max_connections          | 100                                        |
| max_connect_errors       | 10                                         |
| max_delayed_threads      | 20                                         |
| max_heap_table_size      | 16777216                                   |
| max_join_size            | 4294967295                                 |
| max_sort_length          | 1024                                       |
| max_tmp_tables           | 32                                         |
| max_write_lock_count     | 4294967295                                 |
| myisam_recover_options   | OFF                                        |
| myisam_sort_buffer_size  | 8388608                                    |
| net_buffer_length        | 7168                                       |
| net_read_timeout         | 30                                         |
| net_retry_count          | 10                                         |
| net_write_timeout        | 60                                         |
| open_files_limit         | 0                                          |
| pid_file                 | /var/run/mysqld/mysqld.pid                 |
| port                     | 3306                                       |
| protocol_version         | 10                                         |
| record_buffer            | 131072                                     |
| query_buffer_size        | 0                                          |
| safe_show_database       | OFF                                        |
| server_id                | 0                                          |
| skip_locking             | ON                                         |
| skip_networking          | OFF                                        |
| skip_show_database       | OFF                                        |
| slow_launch_time         | 2                                          |
| socket                   | /var/lib/mysql/mysql.sock                  |
| sort_buffer              | 524280                                     |
| table_cache              | 64                                         |
| table_type               | MYISAM                                     |
```

```
| thread_cache_size       | 0                                          |
| thread_stack            | 65536                                      |
| timezone                | EST                                        |
| tmp_table_size          | 1048576                                    |
| tmpdir                  | /tmp/                                      |
| version                 | 3.23.33                                    |
| wait_timeout            | 28800                                      |
--------------------------+--------------------------------------------+
```

From the above table, we can see that the values have been set successfully with the new parameters.

NOTE: It's important to note that the value **key_buffer** cannot be more than 50% of your total memory. Or your system may start to page and become REALLY slow. So, if you have, for example, 256 M of RAM the value can be a maximum of 128 MB and no more.

MySQL Administrative Tools

The commands listed below are some that we use often in normal use, but many more exist and you must check the reference manual for more information.

There are two statements you may use to create new users in the database, the GRANT and INSERT statements. With MySQL you have the possibility to specify, during user creation, what privileges you want to assign to your users. Privileges can be used to set which parts of the database users are allowed to use, administer, control, etc.

The GRANT statement:

The first example below is the steps to follow with the GRANT statements command. In this example we'll create two different users one named "sqladmin" with password "mo" and the second named "operator" with no password and limited privileges.

- To define a new user with a password and full privileges in your database with the GRANT statements, use the following commands:

```
[root@deep /]$ mysql -u root mysql -p
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4 to server version: 3.23.49

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> GRANT ALL PRIVILEGES ON *.* TO sqladmin@localhost
       -> IDENTIFIED BY 'mo' WITH GRANT OPTION;
Query OK, 0 rows affected (0.00 sec)

mysql> \q
Bye
```

The user we have created is called "sqladmin" with the password set to "mo". This user has full privileges "ALL PRIVILEGES" over the database, like the super-user MySQL root. In most cases, we really don't need to create this kind of user for the database.

- To define a new user with limited privileges and no password set with the GRANT statements, use the following commands:

```
[root@deep /]$ mysql -u root mysql -p
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5 to server version: 3.23.49

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> GRANT RELOAD,PROCESS ON *.* TO operator@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql> \q
Bye
```

This second user is called "operator" and is granted the RELOAD and PROCESS administrative privileges only. He doesn't have a password set and can connect from only the localhost without a password. Using the GRANT statement could penalize the performance of the SQL server; it is better to use the INSERT statement, which performs the same function.

The INSERT statement:

The INSERT statements are the second method to create new users for the database. It's interesting to learn this method, since many third party programs use it during user creation. In the example below, we use the same user names as above to show you the differences between both methods.

- To define a new user with password and full privileges in your database with the INSERT statements, use the following commands:

```
[root@deep /]$ mysql -u root mysql -p
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6 to server version: 3.23.49

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> INSERT INTO user VALUES('localhost','sqladmin',PASSWORD('mo'),
    -> 'Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y');
Query OK, 1 row affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> \q
Bye
```

The 14 'Y' you see in this command, represent the privileges allowed for this user, with MySQL version 3.23.51 there are 14 privileges you may associate for the user, since the example user "sqladmin" has full control over the database, the 14 privileges are set to YES 'Y'.

- To define a new user with limited privileges and no password with the INSERT statements, use the following commands:

```
[root@deep /]$ mysql -u root mysql -p
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 7 to server version: 3.23.49

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> INSERT INTO user SET Host='localhost',User='operator',
       -> Reload_priv='Y', Process_priv='Y';
Query OK, 1 row affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> \q
Bye
```

In this second example we can see that only 2 privileges have been set for the user, the RELOAD and PROCESS privileges. Also, this user has no password set and can connect from only the localhost without the need to specify a password.

Of course if you want to specify a password for this user (always recommended), then all you have to do is to include in the INSERT command the line "Password('mypasswd')," after the "User='operator'," parameter.

The UPDATE & DELETE statement:

These two statements can be used to manage user security access to the database. The first statement allows us to update an existing user password in the SQL database and the second statement lets us remove an existing user from the database.

- To update and change a user password from your database, use the commands:

```
[root@deep /]$ mysql -u root mysql -p
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8 to server version: 3.23.49

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> UPDATE user SET Password=PASSWORD('mypasswd') WHERE user='root';
Query OK, 2 rows affected (0.01 sec)
Rows matched: 2  Changed: 2  Warnings: 0

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> \q
Bye
```

In this example, we update and change the password for the super-user called "root". The value **'mypasswd'** is where you put the new password you want to update (this is the only value you must change in the above command).

- To remove a user password from your database, use the following command:

```
[root@deep /]$ mysql -u root mysql -p
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11 to server version: 3.23.49

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> DELETE FROM user WHERE User = "sqladmin";
Query OK, 1 row affected (0.00 sec)

mysql> \q
Bye
```

In this example, we remove the row in the user table of the database related to the user "sqladmin" and all privileges and the password associated with it.

Basic commands:

Most of you already know how SQL databases, and in our case MySQL, work, but for others, this is the first time. Below, I show you the basic commands for managing a database.

- To create a new database, run the **mysqladmin create dbname** utility program:

```
[root@deep /]$ mysqladmin create addressbook -p
Enter password:
```

or with the MySQL terminal monitor program (mysql)

```
[root@deep /]$ mysql -u root mysql -p
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13 to server version: 3.23.49

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> CREATE DATABASE addressbook;
Query OK, 1 row affected (0.00 sec)

mysql> \q
Bye
```

- To delete a database and all its tables, run the **mysqladmin drop** utility program:

```
[root@deep /]$ mysqladmin drop addressbook -p
Enter password:
Dropping the database is potentially a very bad thing to do.
Any data stored in the database will be destroyed.

Do you really want to drop the 'addressbook' database [y/N] y
Database "addressbook" dropped
```

 or with the MySQL terminal monitor program (mysql)

```
[root@deep /]$ mysql -u root mysql -p
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 15 to server version: 3.23.49

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> DROP DATABASE addressbook;
Query OK, 3 rows affected (0.00 sec)

mysql> \q
Bye
```

- To connect to the new database with the MySQL terminal monitor, use the command:

```
mysql> USE addressbook;
Database changed
mysql>
```

- To create a table named contact with the following values, use the command:

```
mysql> CREATE TABLE contact (FirstName VARCHAR(20),
    -> SecondName VARCHAR(20), Address VARCHAR(80),
    -> WorkPhone VARCHAR(25), HomePhone VARCHAR(25),
    -> MobilePhone VARCHAR(25), Fax VARCHAR(25), Website VARCHAR(20),
    -> Mail VARCHAR(30), Title VARCHAR(20), Description VARCHAR(100));
Query OK, 0 rows affected (0.01 sec)

mysql>
```

- To inspect the new table, use the command:

```
mysql> DESCRIBE contact;
+-------------+-------------+--------+--------+----------+----------+
| Field       | Type        | Null   | Key    | Default  |    Extra |
+-------------+-------------+--------+--------+----------+----------+
| FirstName   | varchar(20) | YES    |        | NULL     |          |
| SecondName  | varchar(20) | YES    |        | NULL     |          |
| Address     | varchar(80) | YES    |        | NULL     |          |
| WorkPhone   | varchar(25) | YES    |        | NULL     |          |
| HomePhone   | varchar(25) | YES    |        | NULL     |          |
| MobilePhone | varchar(25) | YES    |        | NULL     |          |
| Fax         | varchar(25) | YES    |        | NULL     |          |
| Website     | varchar(20) | YES    |        | NULL     |          |
| Mail        | varchar(30) | YES    |        | NULL     |          |
| Title       | varchar(20) | YES    |        | NULL     |          |
| Description | varchar(100)| YES    |        | NULL     |          |
+-------------+-------------+--------+--------+----------+----------+
11 rows in set (0.00 sec)

mysql> \q
Bye
```

The `INSERT` statement:

Once your table has been created, you need to populate it. There is statement you may use to do this. The `INSERT` statement is useful when you want to add new records one at a time. You supply values for each column, in the order in which the columns were listed.

- To add a new record using an `INSERT` statement, use this command:

```
mysql> INSERT INTO contact
    -> VALUES ('Henri','Smith','301, Av Washington','(514) 234 8765',
    -> '(514) 456 3290',NULL,NULL,'www.openna.com','henri@openna.com',
    -> 'WebAdmin',NULL);
Query OK, 1 row affected (0.00 sec)

mysql> \q
Bye
```

- To dump the structure and data from `MySQL` databases and tables for backing up, use the following command:

```
[root@deep /]# mysqldump mysql > mysqldb.sql -p
Enter password:
```

In this example, we dump the whole database, named "`mysql`", into a backup file named "`mysqldb.sql`", which can be used later to restore the original database.

- To restore the structure and data from `MySQL` databases and tables from backup, use the following command:

```
[root@deep /]# mysql -u root mysql < mysqldb.sql -p
Enter password:
```

In this example, we restore the original database we backed up earlier named "`mysql`".

Further documentation

For more details, there are many MySQL manual pages that you could read:

```
$ man isamchk (1)          - Check  and  repair of ISAM tables.
$ man isamlog (1)          - Write info about whats in a nisam log file.
$ man mysql (1)            - Text-based  client for mysqld.
$ man mysql_zap (1)        - A perl script used to kill processes.
$ man mysqlaccess (1)      - Create new users to mysql.
$ man mysqladmin (1)       - Utility for performing administrative operations.
$ man mysqld (1)           - Starts the MySQL server demon.
$ man mysqld_multi (1)     - Used to manage several mysqld processes.
$ man mysqldump (1)        - Text-based  client for dumping or backing up mysql databases.
$ man mysqlshow (1)        - Shows the structure of a mysql database.
$ man perror (1)           - Used to display a description for a system error code.
$ man replace (1)          - Utility program that is used by msql2mysql.
$ man safe_mysqld (1)      - Used to start the mysqld daemon on Unix.
```

CHAPTER 39

PostgreSQL

IN THIS CHAPTER

1. **Compiling - Optimizing & Installing `PostgreSQL`**
2. **Configuring `PostgreSQL`**
3. **Running `PostgreSQL` with `SSL` support**
4. **Securing `PostgreSQL`**
5. **Optimizing `PostgreSQL`**
6. **`PostgreSQL` Administrative Tools**

Linux PostgreSQL

Abstract
PostgreSQL, developed originally in the UC Berkeley Computer Science Department, pioneered many of the object-relational concepts now becoming available in commercial databases. It provides SQL92/SQL3 language support, transaction integrity, and type extensibility.

PostgreSQL is an **O**bject-**R**elational **D**ata**b**ase **M**anagement **S**ystem (ORDBMS) based on POSTGRES, Version 4.2, developed at the University of California at Berkeley Computer Science Department. The POSTGRES project, led by Professor Michael Stonebraker, was sponsored by the **D**efense **A**dvanced **R**esearch **P**rojects **A**gency (DARPA), the **A**rmy **R**esearch **O**ffice (ARO), the **N**ational **S**cience **F**oundation (NSF), and ESL, Inc. It is the most advanced open-source database available anywhere.

If your objective is to run many web applications through a SQL database, I recommend you go with MySQL instead of PostgreSQL, not because MySQL is better than PostgreSQL, but only because most of available web applications for Linux on the Internet are primarily made to run with MySQL and more complete documentation exists. With PostgreSQL, most of the web applications will still work, but you will take more work on your part.

These installation instructions assume
Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, at personal discretion).
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest PostgreSQL version number is 7.2.1

The procedures given in this chapter are likely to work on all Linux platforms, but we have only tested it on OpenNA Linux and Red Hat Linux.

Packages
The following is based on information listed by PostgreSQL as of 2002/06/24. Please regularly check http://www.postgresql.org/ for the latest status. We chose to install the required component from a source file because it provides the facility to fine tune the installation.

Source code is available from:
PostgreSQL Homepage: http://www.postgresql.org/
PostgreSQL Site: 216.126.84.28
You must be sure to download: postgresql-7.2.1.tar.gz

Prerequisites
PostgreSQL requires that the software below is already installed on your system to be able to compile successfully. If this is not the case, you must install it. Please make sure you have this program installed on your machine before you proceed with this chapter.

- ✓ OpenSSL is required to run PostgreSQL with SSL support on your system.

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all the files installed on the system in the eventuality of an update in the future. To solve the problem, it is a good idea to make a list of files on the system before you install PostgreSQL, and then one afterwards, and then compare them using the **diff** utility to find out what files were placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep root]# find /* > PostgreSQL1
  ```
- And the following one after you install the software:
  ```
  [root@deep root]# find /* > PostgreSQL2
  ```
- Then use the following command to get a list of what changed:
  ```
  [root@deep root]# diff PostgreSQL1 PostgreSQL2 > PostgreSQL-Installed
  ```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. In the example above, we use the /root directory of the system to store all generated list files.

Compiling - Optimizing & Installing PostgreSQL

Below are the steps that you must make to configure, compile and optimize the PostgreSQL software before installing it onto your system. First off, we install the program as the user "root" so as to avoid permissioning problems.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- This can be done with the following commands:
  ```
  [root@deep /]# cp postgresql-version.tar.gz /var/tmp/
  [root@deep /]# cd /var/tmp/
  [root@deep tmp]# tar xzpf postgresql-version.tar.gz
  ```

Step 2
In order to check that the version of PostgreSQL, which you are going to install, is an original and unmodified one, use the command described below to check its MD5 hashes checksum.

- To verify the MD5 checksum of PostgreSQL, use the following command:
  ```
  [root@deep tmp]# md5sum postgresql-7.2.1.tar.gz
  ```

This should yield an output similar to this:
```
d075e9c49135899645dff57bc58d6233  postgresql-7.2.1.tar.gz
```

Now check that this checksum is exactly the same as the one available into a file called "postgresql-7.1.2.tar.gz.md5" on the PostgreSQL FTP site: 216.126.84.28

Step 3
We must create a new user account called "postgres" with shell privileges on the system to be the owner of the PostgreSQL database files and daemon.

- To create this special PostgreSQL user on OpenNA Linux, use the following command:
```
[root@deep tmp]# groupadd -g 26 postgres > /dev/null 2>&1 || :
[root@deep tmp]# useradd -c "PostgreSQL Server" -d /var/lib/pgsql -g 26
-s /bin/bash -u 26 postgres > /dev/null 2>&1 || :
```

- To create this special PostgreSQL user on Red Hat Linux, use the following command:
```
[root@deep tmp]# groupadd -g 26 postgres > /dev/null 2>&1 || :
[root@deep tmp]# useradd -u 26 -g 26 -s /bin/bash -M -r -d
/var/lib/pgsql postgres > /dev/null 2>&1 || :
```

The above command will create a real account, with no password, but valid shell access for the user postgres because we need it to connect to the database on the terminal of the server.

Step 4
After that, move into the newly created PostgreSQL source directory and perform the following steps to configure and optimize PostgreSQL for your system.

- To move into the newly created PostgreSQL source directory use the command:
```
[root@deep tmp]# cd postgresql-7.2.1/
```

- To configure and optimize PostgreSQL use the following compilation lines:
```
CFLAGS="-O2 -march=i686 -funroll-loops" \
CXXFLAGS="-O2 -march=i686 -funroll-loops -felide-constructors -fno-
exceptions -fno-rtti" \
./configure \
--prefix=/usr \
--includedir=/usr/include \
--localstatedir=/var \
--docdir=/usr/share/doc \
--sysconfdir=/etc \
--mandir=/usr/share/man \
--disable-shared \
--disable-debug \
--disable-nls \
--enable-syslog \
--without-tcl \
--without-perl \
--without-python \
--without-java \
--with-CXX \
--with-openssl \
--with-pam
```

This tells PostgreSQL to set itself up for this particular configuration setup with:

- Disable shared libraries to improve performance.
- Disable build with debugging symbols to get smaller binaries.
- Disable Native Language Support.
- Enables the PostgreSQL server to use the syslog logging facility.
- Build without Tcl and Tk interfaces support.
- Build without Perl interface and PL/Perl support.
- Build without Python interface module support.
- Build without JDBC interface and Java tools support.
- Build C++ modules (libpq++).
- Build with OpenSSL for encryption support.
- Build with PAM support.

WARNING: There is a performance penalty associated with the use of locale support (--enable-locale), but if you are not in an English-speaking environment you will most likely need this configuration line. This option is not included in our compilation lines as shown above.

Step 5
Now, we must make a list of files on the system before installing the software and one afterwards then compare them using the **diff** utility to find out what files are placed where and finally install PostgreSQL:

```
[root@deep postgresql-7.2.1]# make all
[root@deep postgresql-7.2.1]# cd
[root@deep root]# find /* > PostgreSQL1
[root@deep root]# cd /var/tmp/postgresql-7.2.1/
[root@deep postgresql-7.2.1]# make install
[root@deep postgresql-7.2.1]# cd
[root@deep root]# rm -rf /usr/share/doc/postgresql/
[root@deep root]# mkdir -p /var/lib/pgsql
[root@deep root]# chmod 700 /var/lib/pgsql/
[root@deep root]# chown -R postgres.postgres /var/lib/pgsql/
[root@deep root]# strip /usr/bin/postgres
[root@deep root]# strip /usr/bin/psql
[root@deep root]# find /* > PostgreSQL2
[root@deep root]# diff PostgreSQL1 PostgreSQL2 > PostgreSQL-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

Step 6
Once the compilation, optimization and installation of the software has completed, we can free up some disk space by deleting the program tar archive and the related source directory, since they are no longer needed.

- To delete PostgreSQL and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf postgresql-version/
[root@deep tmp]# rm -f postgresql-version.tar.gz
```

Configuring PostgreSQL

After PostgreSQL has been built and installed successfully on your system, the next step is to configure and customize its configuration files to fit your needs. As you'll see further down, we start our configuration of the SQL server with the initialization file, this is important because PostgreSQL needs to start with this file to create all the other configuration files it needs.

- ✓ /etc/init.d/postgresql: (The PostgreSQL Initialization File)
- ✓ /var/lib/pgsql/data/postgresql.conf: (The PostgreSQL Configuration File)

/etc/init.d/postgresql: The PostgreSQL Initialization File

The /etc/init.d/postgresql script file is responsible for automatically starting and stopping the PostgreSQL server. Loading the postgres daemon as a standalone daemon will eliminate load time and will even reduce swapping since non-library code will be shared.

Please note that the following script is only suitable for Linux operating systems using System V. If your Linux system uses some other method, like BSD, you'll have to adjust the script below to make it work for you.

Step 1
Create the **postgresql** script file (touch /etc/init.d/postgresql) and add the lines:

```
#!/bin/bash

# This shell script takes care of starting and stopping PostgreSQL.
#
# chkconfig: 345 85 15
# description: PostgreSQL is a fast & secure SQL database server.
#
# processname: postmaster
# pidfile: /var/run/postmaster.pid

# Source function library.
. /etc/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Source for additional options if we have them.
if [ -f /etc/sysconfig/postgresql ] ; then
        . /etc/sysconfig/postgresql
fi

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

# If PostgreSQL is not available stop now.
[ -f /usr/bin/postmaster ] || exit 0

# Path to the PostgreSQL binary.
postmaster=/usr/bin/postmaster

# Get function listing for cross-distribution logic.
TYPESET=`typeset -f|grep "declare"`

# Some definition for easy maintenance.
PGVERSION=7.2
PGDATA=/var/lib/pgsql/data
```

```
I18N=/etc/sysconfig/i18n

RETVAL=0
prog="PostgreSQL"

start(){
        PSQL_START=$"Starting $prog: "
        echo -n $"Initializing database: "
        if [ ! -d $PGDATA ]
            then
            mkdir -p $PGDATA
            chown postgres.postgres $PGDATA
        fi

        [ -f $I18N ] && cp $I18N $PGDATA/../initdb.i18n
        [ ! -f $I18N ] && echo "LANG=en_US" > $PGDATA/../initdb.i18n

        # Here we initialize the db if not available.
        su -l postgres -s /bin/sh -c "/usr/bin/initdb \
        --pgdata=/var/lib/pgsql/data > /dev/null 2>&1" < /dev/null
        [ -f $PGDATA/PG_VERSION ] && echo_success
        [ ! -f $PGDATA/PG_VERSION ] && echo_failure
        echo

        # Check for postmaster already running...
        pid=`pidof -s postmaster`
        if [ $pid ]
        then
            echo $"Postmaster already running."
        else
        rm -f /tmp/.s.PGSQL.* > /dev/null
        echo -n "$PSQL_START"

        # Here we start PostgreSQL on the server.
        su -l postgres -s /bin/sh -c "/usr/bin/pg_ctl -D \
        $PGDATA -p /usr/bin/postmaster start  > /dev/null 2>&1" < /dev/null
        sleep 1
        pid=`pidof -s postmaster`
        if [ $pid ]
            then
        if echo "$TYPESET"|grep "declare -f success ()" >/dev/null
            then
                success "$PSQL_START"
        else
                echo "  [ OK ]"
        fi
                touch /var/lock/subsys/postgresql
                echo $pid > /var/run/postmaster.pid
                echo
                else
                if echo "$TYPESET"|grep "declare -f failure ()" >/dev/null
                then
                        failure "$PSQL_START"
                        else
                        echo " [ FAILED ]"
                        fi
                        echo
                fi
        fi
}

stop() {
        echo -n $"Shutting down $prog: "
```

```
        su -l postgres -s /bin/sh -c "/usr/bin/pg_ctl stop -D \
        /var/lib/pgsql/data -s -m fast" > /dev/null 2>&1
        ret=$?
        if [ $ret -eq 0 ]; then
            echo_success
        else
            echo_failure
        fi
        echo
        rm -f /var/run/postmaster.pid
        rm -f /var/lock/subsys/postgresql
}

# See how we were called.
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  status)
        status $postmaster
        ;;
  reload)
        su -l postgres -s /bin/sh -c "/usr/bin/pg_ctl reload -D \
        /var/lib/pgsql/data -s" > /dev/null 2>&1
        RETVAL=$?
        ;;
  restart)
        stop
        start
        RETVAL=$?
        ;;
  condrestart)
        if [ -f /var/lock/subsys/postgresql ]; then
                stop
                start
                RETVAL=$?
        fi
        ;;
  *)
        echo $"Usage: $0 {start|stop|status|reload|restart|condrestart}"
        exit 1
esac
exit $?
```

Step 2
Once the postgresql script file has been created, it is important to make it executable, change its default permissions, create the necessary links and start it. Making this file executable will allow the system to run it, changing its default permission is to allow only the root user to change this file for security reasons, and creation of the symbolic links will let the process control initialization of Linux to start the program automatically for you at each boot.

- To make this script executable and to change its default permissions, use the commands:
```
[root@deep /]# chmod 700 /etc/init.d/postgresql
[root@deep /]# chown 0.0 /etc/init.d/postgresql
```

- To create the symbolic rc.d links for PostgreSQL, use the following commands:
```
[root@deep /]# chkconfig --add postgresql
[root@deep /]# chkconfig --level 345 postgresql on
```

- To start PostgreSQL software manually, use the following command:
```
[root@deep /]# /etc/init.d/postgresql start
Initializing database:        [OK]
Starting PostgreSQL:          [OK]
```

Step 3
Once the SQL server has been started, it's time to verify that it is working. With the PostgreSQL server default installation, the only user capable of connecting to the database is the user we have created previously to handle the database files and daemons called "postgres".

- To connect to the PostgreSQL database, perform the following actions:
```
[root@deep /]# psql template1 -U postgres
Welcome to psql, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit

template1=# \q
```

As you can see in the above example, we su to the user called "postgres" before connecting to the database named "template1" through the interactive terminal program "psql" which allows you to interactively enter, edit, and execute SQL commands.

Step 4
Finally, if the SQL server is running and working, it's time to assign a password to the super-user of this database. With PostgreSQL server, this super-user is by default called postgres and has no password assigned to it, which means that anyone could connect with this name and do anything to the database.

- To specify a password for the PostgreSQL super-user, perform the following actions:

```
[root@deep /]# psql template1 -U postgres
Welcome to psql, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit

template1=# ALTER USER postgres WITH PASSWORD 'mypasswd';
ALTER USER
template1=# \q
```

The value **'mypasswd'** as shown above is where you put the password you want to assign for the postgres super-user (this is the only value you must change in the above command).

/var/lib/pgsql/data/postgresql.conf: The PostgreSQL Config File

The /var/lib/pgsql/data/postgresql.conf file is used to specify PostgreSQL system configuration information. Once the SQL server is started, we can reconfigure it to better fit our requirements and run PostgreSQL with improved performance. PostgreSQL automatically creates this file when you start its daemon. Therefore we just have to edit and configure the file.

Step 1
This file is checked each time the database starts its daemon to get the required information. It is also used to specify optimization parameters for the database.

- Edit the **postgresql.conf** file (vi /var/lib/pgsql/data/postgresql.conf) and add/change the following lines inside the file:

```
fsync = false
max_connections = 512
shared_buffers = 1024
silent_mode = true
syslog = 2
log_connections = true
log_timestamp = true
ssl = false
tcpip_socket = false
```

This tells the postgresql.conf file to set itself up for this particular configuration with:

fsync = false
This option "fsync" if set to "false" allows the operating system to do its best in buffering, sorting, and delaying writes, which can make for a considerable performance increase. If you trust your Linux operating system, your hardware and UPS, you can disable this option safety, otherwise enable it. This is a performance feature.

`max_connections = 512`
This option "`max_connections`" determines how many concurrent connections the database server will allow. There is also a compiled-in hard upper limit on this value, which is typically 1024. We increase the default value of "32" to become 512.

`shared_buffers = 1024`
This option "`shared_buffers`" determines the number of shared memory buffers the database server will use. Typically, the integer must be two times (2*) the value of "`max_connections`" parameter, which become in our configuration "`1024`" (2*512=1024). This is a performance feature.

`silent_mode = true`
This option "`silent_mode`" if set to "`true`" will automatically runs `postmaster` in the background and any controlling `ttys` will be disassociated, thus no messages are written to `stdout` or `stderr`. Since we use the `syslog` program on our system to report error messages, we can safety disable this option.

`syslog = 2`
This option "`syslog`" if set to "`2`" will enable the use of `syslog` for logging and will send its output only to `syslog` on the system (`/var/log/messages`).

`log_connections = true`
This option "`log_connections`" if set to "`true`" prints a line about each successful connection to the server log. This is a security feature.

`log_timestamp = true`
This option "`log_timestamp`" if set to "`true`" prefixes each server log message with a timestamp. It's good idea to enable it. This is a security feature.

`ssl = false`
This option "`ssl`", if set to "`true`", enables an `SSL` connection for this `PostgreSQL` server. See later for more information about using `SSL` with `PostgreSQL` and how to use it if you require it. In our configuration, we disable this feature because you have to create the required certificates before enabling this option into your configuration file. If you enable this option now and you do not have the required certificates created and placed in the appropriated location on your server, the `SQL` server will refuse to start and will generate error messages. Therefore, see the section of this chapter relating to `SSL` support with `PostgreSQL` before enabling this parameter.

`tcpip_socket = false`
This option "`tcpip_socket`", if set to "`false`", will accept only local Unix domain socket connections. If you want to allow external connections to your `PostgeSQL` server, then you must change the default value of "`false`" to become "`true`" and see later in this chapter what this implies and how to secure and control external connections. This is a security feature.

Running `PostgreSQL` with `SSL` support

This section applies only if you want to run `PostgreSQL` through an `SSL` connection. Below I'll show you how to set up a certificate to use with `PostgreSQL`. As you can imagine, the principle is the same as for creating a certificate for a web server.

Step 1
First you have to know the **F**ully **Q**ualified **D**omain **N**ame (`FQDN`) of the `PostgreSQL` (`SQL`) Server for which you want to request a certificate. When you want to access your database Server through `sql.domain.com` then the `FQDN` of your `SQL` Server is `sql.domain.com`.

Step 2
Second, select five large and relatively random files from your hard drive (compressed log files are a good start) and put them under your /usr/share/ssl directory. These will act as your random seed enhancers. We refer to them as random1: random2:...: random5 below.

- To select five random files and put them under /usr/share/ssl, use the commands:

```
[root@deep /]# cp /var/log/boot.log /usr/share/ssl/random1
[root@deep /]# cp /var/log/cron /usr/share/ssl/random2
[root@deep /]# cp /var/log/dmesg /usr/share/ssl/random3
[root@deep /]# cp /var/log/messages /usr/share/ssl/random4
[root@deep /]# cp /var/log/secure /usr/share/ssl/random5
```

Step 3
Third, create the RSA private key **not protected with a pass-phrase** for the PostgreSQL Server (it is important to create a RSA private key **without** a pass-phrase, since the PostgreSQL Server cannot ask you during start-up to enter the pass-phrase). The command below will generate 1024 bit RSA Private Key and stores it in the file server.key.

- To generate the Key, use the following commands:

```
[root@deep /]# cd /usr/share/ssl/
[root@deep ssl]# openssl genrsa -rand
random1:random2:random3:random4:random5 -out server.key 1024
123600 semi-random bytes loaded
Generating RSA private key, 1024 bit long modulus
.....................+++++
.....+++++
e is 65537 (0x10001)
```

WARNING: Please backup your **server.key** file. A good choice is to backup this information onto a diskette or other removable media.

Step 4
Finally, generate a **C**ertificate **S**igning **R**equest (CSR) with the server RSA private key. The command below will prompt you for the X.509 attributes of your certificate. Remember to give a name like sql.domain.com when prompted for '**Common Name**'. Do not enter your personal name here. We are requesting a certificate for a Database SQL Server, so the **Common Name** has to match the FQDN of your site.

- To generate the CSR, use the following command:

```
[root@deep ssl]# openssl req -new -key server.key -out server.csr
Using configuration from /usr/share/ssl/openssl.cnf
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [CA]:
State or Province Name (full name) [Quebec]:
Locality Name (eg, city) [Montreal]:
```

```
Organization Name (eg, company) [OpenNA, Inc.]:
Organizational Unit Name (eg, section) [OpenNA, Inc. SQL Server]:
Common Name (eg, YOUR name) [sql.openna.com]:
Email Address [noc@openna.com]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:.
An optional company name []:.
```

WARNING: Make sure you enter the FQDN (**F**ully **Q**ualified **D**omain **N**ame) of the server when OpenSSL prompts you for the "**Common Name**" (i.e. when you generate a CSR for a Database Server which will be later accessed via sql.domain.com, enter sql.domain.com here).

After generation of your **C**ertificate **S**igning **R**equest (CSR), you could send this certificate to a commercial **C**ertifying **A**uthority (CA) like Thawte or Verisign for signing. You usually have to post the CSR into a web form, pay for the signing, await the signed Certificate and store it in an server.crt file. The result is then a real Certificate, which can be used for PostgreSQL.

Step 5
You are not obligated to send your **C**ertificate **S**igning **R**equest (CSR) to a commercial **C**ertifying **A**uthority (CA) for signing. In some cases, and with PostgreSQL Server, you can become your own **C**ertifying **A**uthority (CA) and sign your certificate for yourself.

In the step below, I assume that your CA key pair, which are required for signing certificate by yourself already exist on the server, if this is not the case, please refer to the chapter related to OpenSSL in this book for more information about how to create your CA keys pair and become your own **C**ertifying **A**uthority (CA).

- To sign server CSR's in order to create real SSL Certificates, use the following command:

```
[root@deep ssl]# /usr/share/ssl/misc/sign server.csr
CA signing: server.csr -> server.crt:
Using configuration from ca.config
Enter PEM pass phrase:
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName           :PRINTABLE:'CA'
stateOrProvinceName   :PRINTABLE:'Quebec'
localityName          :PRINTABLE:'Montreal'
organizationName      :PRINTABLE:'OpenNA, Inc.'
organizationalUnitName:PRINTABLE:'OpenNA, Inc. SQL Server'
commonName            :PRINTABLE:'sql.openna.com'
emailAddress          :IA5STRING:'noc@openna.com'
Certificate is to be certified until May 31 13:51:17 2003 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
CA verifying: server.crt <-> CA cert
server.crt: OK
```

This signs the CSR and results in a **server.crt** file.

Step 6
Now, we must place the certificates files (server.key and server.crt) in the data directory of PostgreSQL (/var/lib/pgsql/data) and change their default permission modes to be (0400/-r--------), owned by the user called 'postgres' for PostgreSQL to be able to find and use them when it will start its daemon.

- To place the certificates into the appropriate directory, use the following commands:

```
[root@deep ssl]# mv server.key /var/lib/pgsql/data/
[root@deep ssl]# mv server.crt /var/lib/pgsql/data/
[root@deep ssl]# chmod 400 /var/lib/pgsql/data/server.key
[root@deep ssl]# chmod 400 /var/lib/pgsql/data/server.crt
[root@deep ssl]# chown postgres.postgres /var/lib/pgsql/data/server.key
[root@deep ssl]# chown postgres.postgres /var/lib/pgsql/data/server.crt
[root@deep ssl]# rm -f server.csr
```

First we move both the server.key and server.crt files to the data directory of PostgreSQL. After that we change the permissions and ownership of both certificates to be only readable and owned by the PostgreSQL user called 'postgres' for security reasons. Finally we remove the server.csr file from our system since it is no longer needed.

Step 7
To allow SSL-enabled connections with PostgreSQL, we must change one parameter in the postgresql.conf file.

- Edit the **postgresql.conf** file (vi /var/lib/pgsql/data/postgresql.conf), and change the following line:

```
ssl = false
```

To read:

```
ssl = true
```

Step 8
Next, we have to change the way the PostgreSQL initialization script file should start the SQL server. This is important because starting PostgreSQL with SSL support is different from what we use to start it without SSL support. Here we use the same initialization file we have created previously and change some lines to make it work with SSL support. Text in bold is what we have changed from the default file.

Edit the **postgresql** script file (vi /etc/init.d/postgresql) and change the lines:

```
#!/bin/bash

# This shell script takes care of starting and stopping PostgreSQL.
#
# chkconfig: 345 85 15
# description: PostgreSQL is a fast & secure SQL database server.
#
# processname: postmaster
# pidfile: /var/run/postmaster.pid

# Source function library.
```

```
. /etc/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Source for additional options if we have them.
if [ -f /etc/sysconfig/postgresql ] ; then
        . /etc/sysconfig/postgresql
fi

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

# If PostgreSQL is not available stop now.
[ -f /usr/bin/postmaster ] || exit 0

# Path to the PostgreSQL binary.
postmaster=/usr/bin/postmaster

# Get function listing for cross-distribution logic.
TYPESET=`typeset -f|grep "declare"`

# Some definition for easy maintenance.
PGVERSION=7.2
PGDATA=/var/lib/pgsql/data
I18N=/etc/sysconfig/i18n

RETVAL=0
prog="PostgreSQL"

start(){
        PSQL_START=$"Starting $prog: "
        echo -n $"Initializing database: "
        if [ ! -d $PGDATA ]
            then
            mkdir -p $PGDATA
            chown postgres.postgres $PGDATA
        fi

        [ -f $I18N ] && cp $I18N $PGDATA/../initdb.i18n
        [ ! -f $I18N ] && echo "LANG=en_US" > $PGDATA/../initdb.i18n

        # Here we initialize the db if not available.
        su -l postgres -s /bin/sh -c "/usr/bin/initdb \
        --pgdata=/var/lib/pgsql/data > /dev/null 2>&1" < /dev/null
        [ -f $PGDATA/PG_VERSION ] && echo_success
        [ ! -f $PGDATA/PG_VERSION ] && echo_failure
        echo

        # Check for postmaster already running...
        pid=`pidof -s postmaster`
        if [ $pid ]
        then
            echo $"Postmaster already running."
        else
        rm -f /tmp/.s.PGSQL.* > /dev/null
        echo -n "$PSQL_START"

        # Here we start PostgreSQL with SSL support on the server.
        su -l postgres -s /bin/sh -c -p "/usr/bin/postmaster -D $PGDATA -i -l"
        sleep 1
        pid=`pidof -s postmaster`
        if [ $pid ]
```

```
            then
        if echo "$TYPESET"|grep "declare -f success ()" >/dev/null
            then
                success "$PSQL_START"
        else
                echo "  [ OK ]"
        fi
                touch /var/lock/subsys/postgresql
                echo $pid > /var/run/postmaster.pid
                echo
                else
                if echo "$TYPESET"|grep "declare -f failure ()" >/dev/null
                then
                        failure "$PSQL_START"
                        else
                        echo " [ FAILED ]"
                        fi
                        echo
                fi
        fi
}

stop() {
        echo -n $"Shutting down $prog: "
        su -l postgres -s /bin/sh -c "/usr/bin/pg_ctl stop -D \
        /var/lib/pgsql/data -s -m fast" > /dev/null 2>&1
        ret=$?
        if [ $ret -eq 0 ]; then
            echo_success
        else
            echo_failure
        fi
        echo
        rm -f /var/run/postmaster.pid
        rm -f /var/lock/subsys/postgresql
}

# See how we were called.
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  status)
        status $postmaster
        ;;
  reload)
        su -l postgres -s /bin/sh -c "/usr/bin/pg_ctl reload -D \
        /var/lib/pgsql/data -s" > /dev/null 2>&1
        RETVAL=$?
        ;;
  restart)
        stop
        start
        RETVAL=$?
        ;;
  condrestart)
        if [ -f /var/lock/subsys/postgresql ]; then
                stop
                start
                RETVAL=$?
```

```
        fi
        ;;
  *)
        echo $"Usage: $0 {start|stop|status|reload|restart|condrestart}"
        exit 1
esac
exit $?
```

Step 9
Finally, we must restart our PostgreSQL server for the changes to take effect.

- To restart PostgreSQL use the following command:
```
[root@deep /]# /etc/init.d/postgresql restart
Shutting down PostgreSQL:          [OK]
Initializing database:             [OK]
Starting PostgreSQL:               [OK]
```

Securing PostgreSQL

This section deals with the actions we can make to improve and tighten security with the PostgreSQL database. The interesting point here is that we refer to the features available within the base installed program and not to any additional software.

The PostgreSQL Host-Based Access Control File:

PostgreSQL contains a file named pg_hba.conf located under the /var/lib/pgsql/data directory. The meaning of this file is to control who can connect to each database on the server. Once you look into this file, you'll inevitably remark that connections from clients can be made using a so-called **Unix domain socket** or **Internet domain socket** (i.e. TCP/IP).

A Unix domain socket is when a connection to the database appears from the locahost and an Internet domain socket, as its name implies, is when a connection to the database comes externally (i.e. the Internet) by default all connections from a client to the database server are allowed via the local Unix socket only, not via TCP/IP sockets and the backend must be started with the "tcpip_socket" option set to "true" in the postgresql.conf file to allow non-local clients to connect.

Below, I give some examples for the configuration of the Host-Based Access Control File of PostgreSQL for Unix domain sockets and Internet domain sockets.

Unix domain sockets:

Connections made using Unix domain sockets are controlled as follows by the `pg_hba.conf` file:

```
local DBNAME AUTHTYPE
```

Where **DBNAME** specifies the database this record applies to. The value "`all`" specifies that it applies to all databases and the value "`sameuser`" restricts a user's access to a database with the same user name.

AUTHTYPE specifies the authentication method a user must use to authenticate them selves when connecting to that database. The important different available methods are:

1) **trust** which means that a connection is allowed unconditionally.

2) **reject** which means that a connection is rejected unconditionally.

3) **crypt** which means that the client is asked for a password for the user. This is sent encrypted and is compared against the password held in the `pg_shadow` system catalog table and, if the passwords match then the connection is allowed.

4) **password** which means that the client is asked for a password for the user. This is sent in clear text and compared against the password held in the `pg_shadow` system catalog table again, if the passwords match, the connection is allowed.

Step 1
Now let's see a working example:

- Edit the **pg_hba.conf** file (`vi /var/lib/pgsql/data/pg_hba.conf`), and change the following lines at the end of the file:

```
# By default, allow anything over UNIX domain sockets and localhost.
local        all                                          trust
host         all         127.0.0.1     255.255.255.255     trust
```

To read:

```
# By default, allow anything over UNIX domain sockets and localhost
# only if the user's password in pg_shadow is supplied.
local        all                                          crypt
host         all         127.0.0.1     255.255.255.255     crypt
```

In the above example, we allow all users from UNIX domain sockets and the localhost to connect to all databases, if the user's password in the `pg_shadow` system catalog table is supplied. Recall that user passwords are optionally assigned when a user is created; therefore verify if your users have passwords assigned to them before setting this option.

Step 2
Once the necessary modifications have been set into the `pg_hba.conf` file, it is time to verify if the access control security has been applied to the database.

- Connect to the database called `template1`, by using the following command:

```
[root@deep /]# psql template1 -U postgres
Password:
Welcome to psql, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit

template1=# \q
```

If the system asks you to enter a password, congratulations!

Internet domain sockets:

Connections made using Internet domain sockets are controlled as in the `pg_hba.conf` file:

```
host DBNAME IP_ADDRESS ADDRESS_MASK AUTHTYPE
```

The format is the same as that of the "**`local`**" record type, except that the **`IP_ADDRESS`** and **`ADDRESS_MASK`** are added. **`IP_ADDRESS`** and **`ADDRESS_MASK`** are in the standard dotted decimal `IP` address and mask to identify a set of hosts. These hosts are allowed to connect to the database **`DBNAME`** if the values match.

Step 1
Now see, a working example:

- Edit the **`pg_hba.conf`** file (`vi /var/lib/pgsql/data/pg_hba.conf`), and change the following lines at the end of the file:

```
# By default, allow anything over UNIX domain sockets and localhost
# only if the user's password in pg_shadow is supplied.
local        all                                          crypt
host         all         127.0.0.1     255.255.255.255     crypt
```

To read:

```
# By default, allow anything over UNIX domain sockets and localhost
# only if the user's password in pg_shadow is supplied.
local        all                                          crypt
host         all         127.0.0.1     255.255.255.255     crypt
host         all         0.0.0.0       0.0.0.0             reject
host         all         207.35.78.0   255.255.255.0       crypt
```

In the above example, we kept our previous setting, which allows all users using UNIX domain sockets and localhost to connect to all databases, if the user's password in the `pg_shadow` system catalog table is supplied.

But we have added two new lines, related to the Internet domain sockets that say deny anyone from everywhere, except from any host with IP address 207.35.78.x to make a connection to all databases, unless the user's password in the pg_shadow system catalog table is supplied.

Recall that user passwords are optionally assigned when a user is created; therefore verify that your users passwords have been assigned to them before setting this option.

NOTE: Note that a "host" record will allow regular connections and SSL together. If you want to accept only SSL-secured connections from this host or hosts, you must change every "host" record to become "hostssl" in your pg_hba.conf file.

Step 2
Remember that by default all connections from a client to the database server are only allowed via the local Unix socket, therefore it is important to allow traffic through the PostgreSQL port 5432 in our firewall script file for the database to accept an external connection.

Another important fact is that the backend must be started with the "tcpip_socket" option set to "true" in the postgresql.conf file to allow non-local clients to connect.

- Edit the **postgresql.conf** file (vi /var/lib/pgsql/data/postgresql.conf) and change the following line:

```
fsync = false
max_connections = 512
shared_buffers = 1024
silent_mode = true
syslog = 2
log_connections = true
log_timestamp = true
ssl = false
tcpip_socket = false
```

To read:

```
fsync = false
max_connections = 512
shared_buffers = 1024
silent_mode = true
syslog = 2
log_connections = true
log_timestamp = true
ssl = false
tcpip_socket = true
```

Step 3
Once the required modifications have been made, it is time to verify if the access control security modifications have been applied to the database from the external connection.

- Connect to the database, called `template1`, from another machine, by using:

```
[root@ullyse /]# psql –h 207.35.78.9 template1 -U postgres
Password:
Welcome to psql, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit

template1=# \q
```

If the system asks you to enter a password, congratulations!

Optimizing PostgreSQL

This section deals with actions we can make to improve and tighten performance of PostgreSQL database. Note that we refer to the features available within the base installed program.

Get some fast SCSI hard disk:

One of the most important parts of optimizing PostgreSQL server, as well as for the majority of all SQL databases, is the speed of your hard disk, the faster it is, and the faster your database will run. Consider a SCSI disk with low seeks times like 4.2ms; this can make all the difference, even greater performance can be made with RAID technology.

Skip the updating of the last access time:

The **noatime** attribute of Linux eliminates the need, by the system, to make writes to the file system for files. Mounting the file system where your PostgreSQL databases live with the noatime attribute will avoid some disk seeks and will improve the performance of you SQL server.

If you want to mount the file system of the PostgreSQL database with the noatime attribute, it's important to create and install the PostgreSQL databases in this partition. In our example, we have created this partition early in the chapter 2, this partition is located on /var/lib.

Step 1
To mount the file system of PostgreSQL databases with the noatime option, you must edit the **fstab** file (vi /etc/fstab) and add, to the line that refers to the /var/lib file system, the noatime option after the defaults options as shown below:

- Edit the **fstab** file (vi /etc/fstab), and change the line:

```
LABEL=/var/lib    /var/lib        ext3    defaults        1 2
```

To read:

```
LABEL=/var/lib    /var/lib        ext3    defaults,noatime        1 2
```

NOTE: The line related to `/var/lib` into your `/etc/fstab` file could be different from the one I show above, this is just an example.

Step 2
Once you have made the necessary adjustments to the `/etc/fstab` file, it is time to inform the Linux system about the modifications.

- This can be accomplished with the following commands:
 `[root@deep /]# mount /var/lib -oremount`

Each file system that has been modified must be remounted with the command as shown above.

Step 3
After your file system has been remounted, it is important to verify that the modification of the `fstab` file has been correctly applied.

- You can verify if the modification has been correctly applied with the following command:

```
[root@deep /]# cat /proc/mounts
/dev/root         /           ext3    rw 0 0
/proc             /proc       proc    rw 0 0
/dev/sda1         /boot       ext3    rw 0 0
/dev/sda10        /cache      ext3    rw 0 0
/dev/sda9         /chroot     ext3    rw 0 0
/dev/sda8         /home       ext3    rw 0 0
/dev/sda13        /tmp        ext3    rw 0 0
/dev/sda7         /usr        ext3    rw 0 0
/dev/sda11        /var        ext3    rw 0 0
/dev/sda12        /var/lib    ext3    rw,noatime 0 0
none              /dev/pts    devpts  rw 0 0
```

This command will show you all the file systems on your Linux server and the parameters applied to them. If you see something like:

```
/dev/sda12     /var/lib       ext3       rw,noatime 0 0
```

Congratulations!

NOTE: Look under chapter related to Linux Kernel for more information about the `noatime` attribute and other tunable parameters.

PostgreSQL Administrative Tools

The commands listed below are some that we use often but many more exist and you must check the reference manual for more information.

With `PostgreSQL` Server, passwords can be managed with the query language commands `CREATE USER` and `ALTER USER`, it can also be managed with shell script wrappers around the `SQL` command called `createuser` and `dropuser`. By default, if no password has been set up, the stored password is `NULL` and password authentication will always fail for that user.

The `CREATE USER` query language command:

The first example below is the steps to follow with the `CREATE USER` query language command. In this example we'll create one user called "`sqladmin`" with no password and limited privileges.

- To create a new user in your `PostgreSQL` server with no password and limited privileges, use the following commands:

```
[root@deep /]# psql template1 -U postgres
Password:
Welcome to psql, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit

template1=# CREATE USER sqladmin;
CREATE USER
template1=# \q
```

Since we have not specified any additional clauses to the above query language command, the default clauses will be to deny the new added user the ability to create both databases and new users.

- To create a new user in your `PostgreSQL` server with the password "`mo`" and privileges to create databases and new users, use the following commands:

```
[root@deep /]# psql template1 -U postgres
Password:
Welcome to psql, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit

template1=# CREATE USER sqladmin WITH PASSWORD 'mo' CREATEDB CREATEUSER;
CREATE USER
template1=# \q
```

The ALTER USER query language command:

The ALTER USER query language command can be used to modify user account information on the database. It is important to note that only a database super-user can change privileges and password expiration with this command. Ordinary users can only change their own password.

- To modifies a user account in your PostgreSQL server, use the following commands:

```
[root@deep /]# psql template1 -U postgres
Password:
Welcome to psql, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit

template1=# ALTER USER sqladmin WITH PASSWORD 'mi' NOCREATEUSER;
CREATE USER
template1=# \q
```

In the above example, we modify password for the user sqladmin to become "mi" instead of "mo" and deny him the possibility to created new users by himself.

The shell scripts wrapper createuser and dropuser:

The shell script wrapper creteuser command is the second method to create new users for the database. It's interesting to know this method too, since many third party programs use it during user creation. In the example below, we use the same users name as above to show you the differences between the both methods.

- To create a new user called sqladmin in your PostgreSQL database with no password and privileges to create databases and new users, use the commands:

```
[root@deep /]# su postgres
bash-2.05a$ createuser
Enter name of user to add: sqladmin
Shall the new user be allowed to create databases? (y/n) y
Shall the new user be allowed to create more new users? (y/n) y
Password:
CREATE USER
bash-2.05a$ exit
exit
```

Here we create a new user with no password set called sqladmin with privileges to create databases and new users.

- To create a new user called sqladmin in your PostgreSQL database with the password "mo" and privileges to create databases but not new users, use the commands:

```
[root@deep /]# su postgres
bash-2.05a$ createuser -P
Enter name of user to add: sqladmin
Enter password for user "sqladmin":
Enter it again:
Shall the new user be allowed to create databases? (y/n) y
Shall the new user be allowed to create more new users? (y/n) n
Password:
CREATE USER
bash-2.05a$ exit
exit
```

- To remove a user called sqladmin in your PostgreSQL database, use the commands:

```
[root@deep /]# su postgres
bash-2.05a$ dropuser
Enter name of user to delete: sqladmin
Password:
DROP USER
bash-2.05a$ exit
exit
```

NOTE: By default, users do not have write access to databases they did not create. All files stored within the database are protected from being read by any account other than the postgres super-user account.

Basic commands:

Most of you already know how SQL databases and in our case PostgreSQL work, but for others, this is the first time. Below, I'll show you the basic commands for managing a database.

- To create a new database called "StoreOpenNA" with PostgreSQL, use the commands:

```
[root@deep /]# su postgres
bash-2.05a$ createdb StoreOpenNA
Password:
CREATE DATABASE
bash-2.05a$ exit
exit
```

- To remove a database called "StoreOpenNA" with PostgreSQL, use the commands:

```
[root@deep /]# su postgres
bash-2.05a$ dropdb StoreOpenNA
Password:
DROP DATABASE
bash-2.05a$ exit
exit
```

- To create a new database called “StoreOpenNA” with the PostgreSQL terminal monitor program (psql), use the following commands:

```
[root@deep /]# psql template1 -U postgres
Password:
Welcome to psql, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit

template1=# CREATE DATABASE StoreOpenNA;
CREATE DATABASE
template1=# \q
```

NOTE: Remember that client connections can be restricted by IP address and/or user name via the “**pg_hba.conf**” file under /var/lib/pgsql/data directory.

Other useful PostgreSQL terminal monitor program a (psql), which allow you to interactively enter, edit, and execute SQL commands are:

- To connect to the new database “StoreOpenNA”, use the following command:

```
[root@deep /]# psql template1 -U postgres
Password:
Welcome to psql, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit

template1=# \c storeopenna
You are now connected to database storeopenna.
storeopenna=# \q
```

- To create a table called “bar” under the database storeopenna, use the command:

```
storeopenna=# CREATE TABLE bar (i int4, c char(16));
CREATE
storeopenna=#
```

- To inspect the new table called “bar”, use the following command:

```
storeopenna=# \d bar
           Table "bar"
 Attribute |   Type   | Modifier
-----------+----------+----------
 i         | integer  |
 c         | char(16) |

storeopenna=# \q
```

Further documentation

For more details, there are many PostgreSQL manual pages that you could read:

```
$ man initdb (1)            - Create a new PostgreSQL database cluster.
$ man pg_passwd (1)         - Change a secondary PostgreSQL password file.
$ man postgres (1)          - Run a PostgreSQL server in single-user mode.
$ man postmaster (1)        - PostgreSQL multiuser database server.
$ man createdb (1)          - Create a new PostgreSQL database.
$ man createlang (1)        - Define a new PostgreSQL procedural language.
$ man createuser (1)        - Define a new PostgreSQL user account.
$ man dropdb (1)            - Remove a PostgreSQL database.
$ man droplang (1)          - Remove a PostgreSQL procedural language.
$ man dropuser (1)          - Remove a PostgreSQL user account.
$ man pg_dump (1)           - Extract a PostgreSQL database into a script file.
$ man pg_dumpall (1)        - Extract all PostgreSQL databases into a script file.
$ man psql (1)              - PostgreSQL interactive terminal.
$ man vacuumdb (1)          - Garbage-collect and analyze a PostgreSQL database.
$ man abort (7)             - Abort the current transaction.
$ man alter_group (7)       - Add users to a group or remove users from a group.
$ man alter_table (7)       - Change the definition of a table.
$ man alter_user (7)        - Change a database user account.
$ man create_database (7)   - Create a new database.
$ man create_operator (7)   - Define a new operator.
$ man create_table (7)      - Define a new table.
$ man create_user (7)       - Define a new database user account.
$ man delete (7)            - Delete rows of a table.
$ man drop_database (7)     - Remove a database.
$ man drop_table (7)        - Remove a table.
$ man drop_user (7)         - Remove a database user account.
$ man grant (7)             - Define access privileges.
$ man insert (7)            - Create new rows in a table.
$ man revoke (7)            - Remove access privileges.
$ man select (7)            - Retrieve rows from a table or view.
$ man show (7)              - Show the value of a run-time parameter.
```

CHAPTER 40

OpenLDAP

IN THIS CHAPTER

1. **Compiling - Optimizing & Installing `OpenLDAP`**
2. **Configuring `OpenLDAP`**
3. **Running `OpenLDAP` with `TLS/SSL` support**
4. **Running `OpenLDAP` in a chroot jail**
5. **Securing `OpenLDAP`**
6. **Optimizing `OpenLDAP`**
7. **`OpenLDAP` Administrative Tools**
8. **`OpenLDAP` Users Tools**

Linux OpenLDAP

Abstract

Until now, we have been talking about security and optimization in this book, so why would we want to talk about OpenLDAP? Well, the OpenLDAP directory server will expand our horizons through its many possibilities. We can use its replication capability to centralize and consolidate different information on one server for all the other servers on our network.

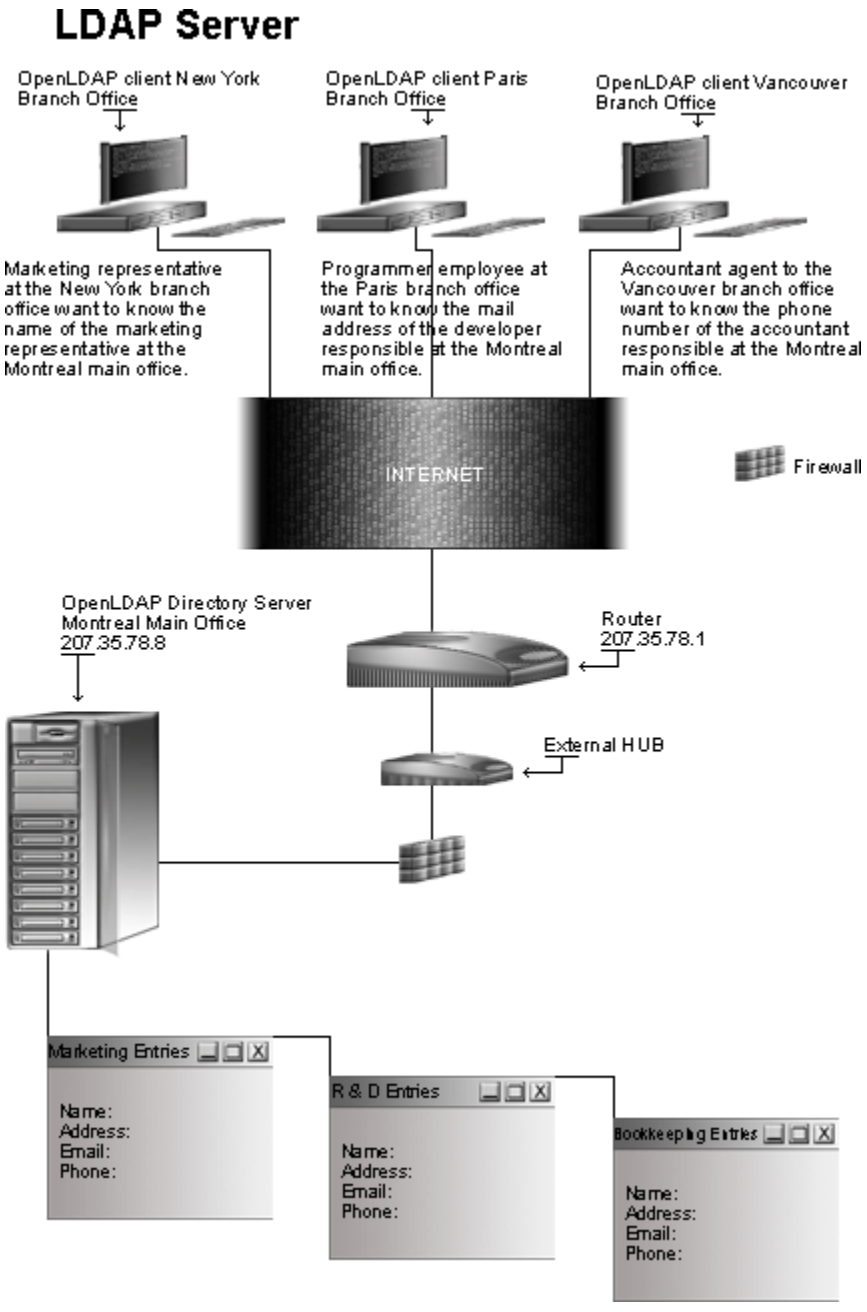
Imagine having the possibility of adding or disabling a UNIX or NT account, setting access to a restricted Web server, and adding a mail address or alias, all with a single operation available as an NIS service, with the added security of SSL encryption, and with the speed of object-oriented hierarchies. Another interesting use is to create an authoritative list of employees on one or more LDAP servers that can be accessible from your private network, or over the Internet.

At present OpenLDAP on Linux is typically used to associate names with phone numbers and e-mail addresses, but in the future this will almost certainly change. Directories are designed to support a high volume of queries since the data in the directory doesn't change all that often, therefore, we can imagine an interesting use of OpenLDAP as a possible Domain Name System alternative, mail server access and control, web server authentication, and many other possibilities.

LDAP (**L**ightweight **D**irectory **A**ccess **P**rotocol) is an open-standard protocol for accessing information services. The protocol runs over Internet transport protocols, such as TCP, and can be used to access stand-alone directory servers or X.500 directories. X.500 is an international standard for full-featured directories, which is complex, requires lots of computing resources and the full OSI stack. LDAP in contrast, can run easily on a PC and use the TCP/IP protocol.

In our installation we'll run OpenLDAP as non root-user and in a chrooted environment with TSL/SSL support. You can configure many different kinds of backend databases with OpenLDAP. A high-performance, disk-based database named "LDBM"; a database interface to arbitrary UNIX commands or shell scripts named "SHELL"; a simple password file database named "PASSWD", and others like SQL.

The default installation of OpenLDAP assumes an LDBM backend database and this is the one that we'll show you in this chapter. For the other types of backend databases, you must add to your configuration the required options.


LDAP Server
OpenLDAP client New York Branch Office
OpenLDAP client Paris Branch Office
OpenLDAP client Vancouver Branch Office
Marketing representative at the New York branch office want to know the name of the marketing representative at the Montreal main office.
Programmer employee at the Paris branch office want to know the mail address of the developer responsible at the Montreal main office.
Accountant agent to the Vancouver branch office want to know the phone number of the accountant responsible at the Montreal main office.
INTERNET
Firewall
OpenLDAP Directory Server Montreal Main Office 207.35.78.8
Router 207.35.78.1
External HUB
Marketing Entries
Name:
Address:
Email:
Phone:
R & D Entries
Name:
Address:
Email:
Phone:
Bookkeeping Entries
Name:
Address:
Email:
Phone:

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, at personal discretion).
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest OpenLDAP version number is 2.1.2

The procedures given in this chapter are likely to work on all Linux platforms, but we have only tested it on OpenNA Linux and Red Hat Linux.

Packages

The following is based on information listed by OpenLDAP as of 2002/06/24. Please regularly check http://www.openldap.org/ for the latest status. We chose to install the required component from a source file because it provides the facility to fine tune the installation.

Source code is available from:
OpenLDAP Homepage: http://www.openldap.org/
OpenLDAP FTP Site: 204.152.186.57
You must be sure to download: openldap-2.1.2.tgz

Prerequisites

OpenLDAP requires that the software below is already installed on your system to be able to compile successfully. If this is not the case, you must install it. Please make sure you have this program installed on your machine before you proceed with this chapter.

- ✓ OpenSSL is required to run OpenLDAP with SSL support on your system.

NOTE: For more information on OpenSSL software, please see earlier chapters in this book.

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all the files installed on the system in the eventuality of an update in the future. To solve the problem, it is a good idea to make a list of files on the system before you install OpenLDAP, and then one afterwards, and then compare them using the **diff** utility to find out what files were placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep root]# find /* > OpenLDAP1
  ```
- And the following one after you install the software:
  ```
  [root@deep root]# find /* > OpenLDAP2
  ```
- Then use the following command to get a list of what changed:
  ```
  [root@deep root]# diff OpenLDAP1 OpenLDAP2 > OpenLDAP-Installed
  ```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. In the example above, we use the /root directory of the system to store all generated list files.

Compiling - Optimizing & Installing OpenLDAP

Below are the steps that you must make to configure, compile and optimize the OpenLDAP **L**ightweight **D**irectory **A**ccess **P**rotocol (LDAP) server software before installing it onto your system. First off, we install the program as the user 'root' so as to avoid permissioning problems.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- This can be done with the following commands:
```
[root@deep /]# cp openldap-version.tgz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf openldap-version.tgz
```

Step 2
In order to check that the version of OpenLDAP, which you are going to install, is an original and unmodified one, use the command described below to check its MD5 hashes checksum.

- To verify the MD5 checksum of OpenLDAP, use the following command:
```
[root@deep tmp]# md5sum openldap-2.1.2.tgz
```

This should yield an output similar to this:
```
bd7dccdfe00850525464c8c9aa119866  openldap-2.1.2.tgz
```

Now check that this checksum is exactly the same as the one available into a file called "openldap-2.1.2.md5" on the OpenLDAP FTP site: 204.152.186.57

Step 3
OpenLDAP needs a UID and GID to properly run on the system but this UID/GID cannot run as super-user root; for this reason we must create a special user with no shell privileges on the system for running OpenLDAP daemon.

- To create this special OpenLDAP user on OpenNA Linux, use the following command:
```
[root@deep tmp]# groupadd -g 55 ldap > /dev/null 2>&1 || :
[root@deep tmp]# useradd -c "OpenLDAP Server" -d /var/lib/ldap -g 55 -s
/bin/false -u 55 ldap > /dev/null 2>&1 || :
```

- To create this special OpenLDAP user on Red Hat Linux, use the following command:
```
[root@deep tmp]# groupadd -g 55 ldap > /dev/null 2>&1 || :
[root@deep tmp]# useradd -u 55 -g 55 -s /bin/false -M -r -d
/var/lib/ldap ldap > /dev/null 2>&1 || :
```

The above command will create a null account, with no password, no valid shell, no files owned-nothing but a UID and a GID for the program. Remember that OpenLDAP daemon does not need to have a shell account on the server.

Step 4
Now, edit the **shells** file (`vi /etc/shells`) and add a non-existent shell name "`/bin/false`", which is the one we used in the `useradd` command above.

```
[root@deep tmp]# vi /etc/shells
/bin/bash2
/bin/bash
/bin/sh
/bin/false ← This is our added no-existent shell
```

Step 5
After that, move into the newly created `OpenLDAP` source directory and perform the following steps to configure and optimize the software for your system.

- To move into the newly created `OpenLDAP` source directory use the command:
 `[root@deep tmp]# cd openldap-2.1.2/`

Step 6
There are some source files to modify before going in configuration and compilation of the program; the changes allow us to fix some problems and file locations. There is lot of changes to make but we need to do it if we want to have a working program.

- Edit the **slap.h** file (`vi +15 servers/slapd/slap.h`) and change the lines:

  ```
  #include <sys/types.h>
  #include <ac/syslog.h>
  #include <ac/regex.h>
  #include <ac/socket.h>
  #include <ac/time.h>
  #include <ac/param.h>
  ```

 To read:

  ```
  #include <sys/types.h>
  #include <sys/socket.h>
  #include <ac/syslog.h>
  #include <ac/regex.h>
  #include <ac/socket.h>
  #include <ac/time.h>
  #include <ac/param.h>
  ```

- Edit the **back-ldbm.h** file (`vi +23 servers/slapd/back-ldbm/back-ldbm.h`) and change the line:

  ```
  #define DEFAULT_DB_DIRECTORY   LDAP_RUNDIR LDAP_DIRSEP "openldap-ldbm"
  ```

 To read:

  ```
  #define DEFAULT_DB_DIRECTORY   "/var/lib/ldap"
  ```

- Edit the **slurp.h** file (`vi +47 servers/slurpd/slurp.h`) and change the line:

```
#define DEFAULT_SLURPD_REPLICA_DIR      LDAP_RUNDIR LDAP_DIRSEP
"openldap-slurp"
```

To read:

```
#define DEFAULT_SLURPD_REPLICA_DIR      "/var/lib/ldap"
```

- Edit the **slurp.h** file (`vi +56 servers/slurpd/slurp.h`) and change the line:

```
#define SLURPD_DUMPFILE                 LDAP_TMPDIR LDAP_DIRSEP
"slurpd.dump"
```

To read:

```
#define SLURPD_DUMPFILE                 DEFAULT_SLURPD_REPLICA_DIR
"/slurpd.dump"
```

- Edit the **mod.mk** file (`vi +13 build/mod.mk`) and change the line:

```
LTFLAGS = --only-$(LINKAGE)
```

To read:

```
#LTFLAGS = --only-$(LINKAGE)
```

- Edit the **top.mk** file (`vi +101 build/top.mk`) and change the line:

```
LDAP_LIBPATH= -L$(LDAP_LIBADIR)
```

To read:

```
LDAP_LIBPATH= -L. -L$(LDAP_LIBADIR)/libavl -L$(LDAP_LIBADIR)/liblber -
L$(LDAP_LIBADIR)/liblber/.libs -L$(LDAP_LIBADIR)/libldap -
L$(LDAP_LIBADIR)/libldap/.libs -L$(LDAP_LIBADIR)/libldap_r -
L$(LDAP_LIBADIR)/libldap_r/.libs -L$(LDAP_LIBADIR)/libldbm -
L$(LDAP_LIBADIR)/libldif -L$(LDAP_LIBADIR)/liblunicode -
L$(LDAP_LIBADIR)/liblutil
```

- Edit the **lib-shared.mk** file (`vi +34 build/lib-shared.mk`) and change the line:

```
$(LTLIBLINK) -o $@ $(OBJS) version.lo $(EXTRA_LIBS)
```

To read:

```
$(LTLIBLINK) -o $@ $(OBJS) version.lo $(EXTRA_LIBS) $(EXTRA_DEPS)
```

- Edit the **Makefile.in** file (`vi +53 libraries/libldap/Makefile.in`) and add/change the lines:

```
EXTRA_DEFS = $(@PLAT@_@LIB_LINKAGE@_LIB_DEFS)
EXTRA_LIBS = $(@PLAT@_@LIB_LINKAGE@_LIB_LIBS) $(@PLAT@_XXLIBS)
```

 To read:

```
EXTRA_DEFS = $(@PLAT@_@LIB_LINKAGE@_LIB_DEFS)
EXTRA_LIBS = $(@PLAT@_@LIB_LINKAGE@_LIB_LIBS) $(@PLAT@_XXLIBS)
EXTRA_DEPS = ../liblber/liblber.la
```

- Edit the **Makefile.in** file (`vi +62 libraries/libldap_r/Makefile.in`) and add/change the lines:

```
EXTRA_DEFS = $(@PLAT@_@LIB_LINKAGE@_LIB_DEFS)
EXTRA_LIBS = $(@PLAT@_@LIB_LINKAGE@_LIB_LIBS) $(@PLAT@_XXLIBS)
```

 To read:

```
EXTRA_DEFS = $(@PLAT@_@LIB_LINKAGE@_LIB_DEFS)
EXTRA_LIBS = $(@PLAT@_@LIB_LINKAGE@_LIB_LIBS) $(@PLAT@_XXLIBS)
EXTRA_DEPS = ../liblber/liblber.la
```

Step 7
Once the modifications have been made to the source files of `OpenLDAP`, it is time configure and optimize it for our system.

- To configure and optimize `OpenLDAP` use the following compilation lines:

```
CFLAGS="-O2 -march=i686 -funroll-loops -D_REENTRANT -fPIC"; export CFLAGS
./configure \
--prefix=/usr \
--libexecdir=/usr/sbin \
--sysconfdir=/etc \
--localstatedir=/var/run \
--mandir=/usr/share/man \
--disable-debug \
--disable-ipv6 \
--disable-local \
--disable-shared \
--enable-syslog \
--enable-crypt \
--enable-static \
--enable-passwd \
--enable-shell \
--enable-ldbm \
--without-threads \
--with-ldbm-api=gdbm \
--with-tls
```

This tells OpenLDAP to set itself up for this particular configuration setup with:

- Disable debugging support to improve performance.
- Disable IPv6 support.
- Disable AF_LOCAL (AF_UNIX) socket support.
- Disable shared libraries support to improve performance.
- Enable syslog support.
- Enable crypt(3) passwords support.
- Enable and build static libraries for better performance.
- Enable passwd backend support with OpenLDAP.
- Enable shell backend support with OpenLDAP.
- Enable ldbm backend support with OpenLDAP.
- Disable threads support for OpenLDAP on the system
- Enable and include TLS/SSL encryption support into the software.

Step 8
Now, we must make a list of all existing files on the system before installing the software, and one afterwards, then compare them using the **diff** utility to find out what files are placed where and finally install OpenLDAP **L**ightweight **D**irectory **A**ccess **P**rotocol (LDAP) server.

```
[root@deep openldap-2.1.2]# make depend
[root@deep openldap-2.1.2]# make
[root@deep openldap-2.1.2]# make test
[root@deep openldap-2.1.2]# cd
[root@deep root]# find /* > OpenLDAP1
[root@deep root]# cd /var/tmp/openldap-2.1.2/
[root@deep openldap-2.1.2]# make install
[root@deep openldap-2.1.2]# mkdir -p -m0700 /var/lib/ldap
[root@deep openldap-2.1.2]# chown -R ldap.ldap /var/lib/ldap/
[root@deep openldap-2.1.2]# rm -rf /var/run/openldap-ldbm/
[root@deep openldap-2.1.2]# rm -rf /var/run/openldap-slurp/
[root@deep openldap-2.1.2]# rm -f /etc/openldap/*.default
[root@deep openldap-2.1.2]# rm -f /etc/openldap/schema/*.default
[root@deep openldap-2.1.2]# cd
[root@deep root]# find /* > OpenLDAP2
[root@deep root]# diff OpenLDAP1 OpenLDAP2 > OpenLDAP-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

Step 9
Once the configuration, optimization, compilation, and installation of the **L**ightweight **D**irectory **A**ccess **P**rotocol (LDAP) server software has been accomplished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete OpenLDAP and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf openldap-version/
[root@deep tmp]# rm -f openldap-version.tgz
```

Configuring OpenLDAP

After OpenLDAP has been built and installed successfully on your system, the next step is to configure and customize its configuration files to fit your needs.

- ✓ /etc/openldap/slapd.conf: (The OpenLDAP Configuration File)
- ✓ /etc/init.d/ldap: (The OpenLDAP Initialization File)

/etc/openldap/slapd.conf: The OpenLDAP Configuration File

The slapd.conf file is the main configuration file for the stand-alone slapd daemon and for all of the database back-ends. Options like: permission, password, database type, database location and so on can be configured in this file and will apply to the "slapd" daemon as a whole.

In the example below we configure the slapd.conf file for an LDBM backend database. Text in bold is the parts of the file that must be customized and adjusted to suit our requirements.

Step 1
The first thing to do before starting your **L**ightweight **D**irectory **A**ccess **P**rotocol (LDAP) server is to edit its slapd.conf file and change the contents to reflect your environment and setup.

- Edit the **slapd.conf** file (vi /etc/openldap/slapd.conf) and add/adjust the following information:

```
# See slapd.conf(5) for details on configuration options.
# This file should NOT be world readable.
include         /etc/openldap/schema/core.schema

# Do not enable referrals until AFTER you have a working directory
# service AND an understanding of referrals.
#referral       ldap://root.openldap.org

pidfile         /var/run/slapd.pid
argsfile        /var/run/slapd.args

# Load dynamic backend modules:
# modulepath    /usr/sbin/openldap
# moduleload    back_ldap.la
# moduleload    back_ldbm.la
# moduleload    back_passwd.la
# moduleload    back_shell.la

# Access Control List:
defaultaccess read
access to attr=userpassword
        by self write
        by dn="cn=Manager,dc=openna,dc=com" write
        by * compare

#######################################################################
# ldbm database definitions
#######################################################################

database        ldbm
readonly        off
suffix          "dc=openna,dc=com"
rootdn          "cn=Manager,dc=openna,dc=com"

# Cleartext passwords, especially for the rootdn, should
# be avoid. See slappasswd(8) and slapd.conf(5) for details.
```

```
# Use of strong authentication encouraged.
rootpw          secret

# The database directory MUST exist prior to running slapd AND
# should only be accessible by the slapd/tools. Mode 700 recommended.
directory       /var/lib/ldap

# Indices to maintain:
index uid       pres,eq
index cn,sn     pres,eq,sub
index objectClass eq
```

This tells the `slapd.conf` file to set itself up for this particular configuration with:

```
defaultaccess read
access to attr=userpassword
    by self write
    by dn="cn=Manager,dc=openna,dc=com" write
    by * compare
```

These directives in the `slapd.conf` file relate to access control in the `LDAP` directory. The access configuration file directive shown above is used to control access to `slapd` daemon entries and attributes in the system.

This example applies to all entries in the "`dc=openna,dc=com`" sub tree and means that read access is granted to everyone <`defaultaccess read`>, and the entry itself can write all attributes, except for *`userpassword`*. The *`userpassword`* attribute is writeable only by the specified `cn` entry (`Manager`), and is comparable by everybody else. See your user manual for more information on these options. This is a security feature.

`readonly  off`
This directive if set to "`on`" puts the database into "`read-only`" mode. Any attempts to modify the database will return an "unwilling to perform" error. It is useful when you make you directory service available to the public. Since we need to populate our directory with information, we will set the directive to "`off`" and change it to "`on`" only if we don't need to add any additional information inside the database. This is a security feature.

`suffix  "dc=openna,dc=com"`
This directive specifies the **D**istinguished **N**ame (`DN`) of the root of the sub tree you are trying to create. In other words, it indicates what entries are to be held by this database. In most cases, we should define our domain name here but depending of the type of directory you wish to run, this may change.

`rootdn  "cn=Manager,dc=openna,dc=com"`
This directive specifies the **D**istinguished **N**ame (`DN`) of the entry allowed to do everything on the `LDAP` directory. This `DN` is not subject to access control or administrative limitations for operations on this database. The name entered here can be one that doesn't actually exist in your password file `/etc/passwd`.

`rootpw  secret`
This directive specifies the password that can be used to authenticate the super-user entry of the database. This is the password for the `DN` given above that will always work, regardless of whether an entry with the given `DN` exists or has a password. It's important to avoid the use of clear text passwords here and to use a crypto password instead. In our example, the password is "`secret`". This is a security feature.

```
directory       /var/lib/ldap
```

This directive specifies the directory where the database and associated index files of LDAP should reside. We must set this to /var/lib/ldap because we created this directory earlier in the installation stage specifically to handle the backend database of LDAP.

```
index uid       pres,eq
index cn,sn     pres,eq,sub
index objectClass eq
```

These directives specify the index definitions you want to build and maintain for the given attribute in the database definition. The options we specify in our slapd.conf example file as shown above, cause all quality indexes to be maintained for the uid, cn, sn and objectClass attributes; an all indexes for the uid, cn and sn attributes. See the OpenLDAP user manual for more information on these options.

Step 2
Once you have set your preferences and environment in the slapd.conf file, it is important to change its default permissions and owner to be the user (ldap) under which the **L**ightweight **D**irectory **A**ccess **P**rotocol (LDAP) server will runs.

- To change the permissions and owner of this file, use the following commands:
  ```
  [root@deep /]# chmod 600 /etc/openldap/slapd.conf
  [root@deep /]# chown ldap.ldap /etc/openldap/slapd.conf
  ```

/etc/init.d/ldap: The OpenLDAP Initialization File

The /etc/init.d/ldap script file is responsible for automatically starting and stopping the OpenLDAP server. Loading the slapd and/or slurpd daemon as a standalone daemon will eliminate load time and will even reduce swapping since non-library code will be shared.

Please note that the following script is only suitable for Linux operating systems using System V. If your Linux system uses some other method, like BSD, you'll have to adjust the script below to make it work for you.

Step 1
Create the **ldap** script file (touch /etc/init.d/ldap) and add the following lines:

```
#!/bin/bash

# This shell script takes care of starting and stopping OpenLDAP.
#
# chkconfig: 345 39 61
# description: LDAP stands for Lightweight Directory Access Protocol, used \
#              for implementing the industry standard directory services.
#
# processname: slapd
# config: /etc/openldap/slapd.conf
# pidfile: /var/run/slapd.pid

# Source function library.
. /etc/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network
```

```
# Source for additional options if we have them.
if [ -f /etc/sysconfig/ldap ] ; then
        . /etc/sysconfig/ldap
fi

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

# If OpenLDAP is not available stop now.
[ -f /usr/sbin/slapd ] || exit 0

# Path to the OpenLDAP binaries.
slapd=/usr/sbin/slapd
slurpd=/usr/sbin/slurpd

RETVAL=0
prog="OpenLDAP"

start() {
        echo -n $"Starting $prog: "
        if grep -q ^TLS /etc/openldap/slapd.conf ; then
        daemon $slapd -u ldap -h '"ldap:/// ldaps:///"'
            RETVAL=$?
        else
        daemon $slapd -u ldap
            RETVAL=$?
        fi
        echo
        if [ $RETVAL -eq 0 ]; then
            if grep -q "^replogfile" /etc/openldap/slapd.conf; then
                echo -n $"Starting $prog: "
                daemon $slurpd
                RETVAL=$?
                echo
            fi
        fi
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/ldap
        return $RETVAL
}

stop() {
        echo -n $"Shutting down $prog: "
        killproc $slapd
        RETVAL=$?
        echo
        if [ $RETVAL -eq 0 ]; then
            if grep -q "^replogfile" /etc/openldap/slapd.conf; then
                echo -n $"Shutting down $prog: "
                killproc $slurpd
                RETVAL=$?
                echo
            fi
        fi
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/ldap /var/run/slapd.args
        return $RETVAL
}

# See how we were called.
case "$1" in
  start)
        start
        ;;
  stop)
```

```
        stop
        ;;
  status)
        status $slapd
        if grep -q "^replogfile" /etc/openldap/slapd.conf ; then
           status $slurpd
        fi
        ;;
  restart)
        stop
        start
        RETVAL=$?
        ;;
  condrestart)
        if [ -f /var/lock/subsys/ldap ] ; then
            stop
            start
            RETVAL=$?
        fi
        ;;
  *)
        echo $"Usage: $0 {start|stop|status|restart|condrestart}"
        exit 1
esac
exit $RETVAL
```

Step 2
Once the `ldap` script file has been created, it is important to make it executable, change its default permissions, create the necessary links and start it. Making this file executable will allow the system to run it, changing its default permission is to allow only the root user to change this file for security reasons, and the creation of the symbolic links will let the process control initialization of Linux start the program automatically for you at each reboot.

- To make this script executable and to change its default permissions, use the commands:
  ```
  [root@deep /]# chmod 700 /etc/init.d/ldap
  [root@deep /]# chown 0.0 /etc/init.d/ldap
  ```

- To create the symbolic `rc.d` links for `OpenLDAP`, use the following commands:
  ```
  [root@deep /]# chkconfig --add ldap
  [root@deep /]# chkconfig --level 345 ldap on
  ```

- To start `OpenLDAP` software manually, use the following command:
  ```
  [root@deep /]# /etc/init.d/ldap start
  Starting OpenLDAP:            [OK]
  ```

Running OpenLDAP with TLS/SSL support

This section applies only if you want to run OpenLDAP through an SSL connection. Below I show you how to set up a certificate for use with OpenLDAP, the principle is the same as for creating a certificate for a Web Server (refer to OpenSSL chapter if you have problems creating the certificates).

Step 1
First you have to know the **F**ully **Q**ualified **D**omain **N**ame (FQDN) of the **L**ightweight **D**irectory **A**ccess **P**rotocol (LDAP) server for which you want to request a certificate. When you want to access your **L**ightweight **D**irectory **A**ccess **P**rotocol (LDAP) server through ldap.domain.com then the FQDN of your OpenLDAP server is ldap.domain.com.

Step 2
Second, select five large and relatively random files from your hard drive (compressed log files are a good start) and put them under your /usr/share/ssl directory. These will act as your random seed enhancers. We refer to them as random1: random2:...: random5 below.

- To select five random files and put them under /usr/share/ssl, use the commands:

```
[root@deep /]# cp /var/log/boot.log /usr/share/ssl/random1
[root@deep /]# cp /var/log/cron /usr/share/ssl/random2
[root@deep /]# cp /var/log/dmesg /usr/share/ssl/random3
[root@deep /]# cp /var/log/messages /usr/share/ssl/random4
[root@deep /]# cp /var/log/secure /usr/share/ssl/random5
```

Step 3
Third, create the RSA private key **not protected with a pass-phrase** for the OpenLDAP server (it is important to create a RSA private key **without** a pass-phrase since the OpenLDAP server cannot ask you during start-up to enter the pass-phrase). The command below will generate a 1024 bit RSA Private Key and stores it in the file ldap.key.

- To generate the Key, use the following commands:

```
[root@deep /]# cd /usr/share/ssl/
[root@deep ssl]# openssl genrsa -rand
random1:random2:random3:random4:random5 -out ldap.key 1024
123600 semi-random bytes loaded
Generating RSA private key, 1024 bit long modulus
.....................+++++
.....+++++
e is 65537 (0x10001)
```

WARNING: Please backup your **ldap.key** file. A good choice is to backup this information onto a diskette or other removable media.

Step 4
Finally, generate a **C**ertificate **S**igning **R**equest (CSR) with the server RSA private key. The command below will prompt you for the X.509 attributes of your certificate. Remember to give the name ldap.domain.com when prompted for '**Common Name**'. Do not enter your personal name here. We are requesting a certificate for a **L**ightweight **D**irectory **A**ccess **P**rotocol (LDAP) server, so the **Common Name** has to match the FQDN of your website.

- To generate the CSR, use the following command:

```
[root@deep ssl]# openssl req -new -key ldap.key -out ldap.csr
Using configuration from /usr/share/ssl/openssl.cnf
Enter PEM pass phrase:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [CA]:
State or Province Name (full name) [Quebec]:
Locality Name (eg, city) [Montreal]:
Organization Name (eg, company) [OpenNA, Inc.]:
Organizational Unit Name (eg, section) [OpenNA, Inc. LDAP Server]:
Common Name (eg, YOUR name) [ldap.openna.com]:
Email Address [noc@openna.com]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:.
An optional company name []:.
```

WARNING: Make sure you enter the FQDN (**F**ully **Q**ualified **D**omain **N**ame) of the server when OpenSSL prompts you for the "**Common Name**" (i.e. when you generate a CSR for a LDAP server which will be later accessed via ldap.domain.com, enter ldap.domain.com here).

After the generation of your **C**ertificate **S**igning **R**equest (CSR), you could send this certificate to a commercial **C**ertifying **A**uthority (CA) like Thawte or Verisign for signing. You usually have to post the CSR into a web form, pay for the signing, await the signed Certificate and store it into an ldap.crt file. The result is then a real Certificate, which can be used for OpenLDAP.

Step 5
You are not obligated to send your **C**ertificate **S**igning **R**equest (CSR) to a commercial **C**ertifying **A**uthority (CA) for signing. In some cases, and with OpenLDAP Directory Server, you can become your own **C**ertifying **A**uthority (CA) and sign your certificate by yourself. In the step below, I assume that your CA keys pair, which is required for signing certificate by yourself, already exists on the server, if this is not the case, please refer to the chapter related to OpenSSL in this book for more information about how to create your CA keys pair and become your own **C**ertifying **A**uthority (CA).

- To sign server CSR's in order to create real SSL Certificates, use the following command:

```
[root@deep ssl]# /usr/share/ssl/misc/sign ldap.csr
CA signing: ldap.csr -> ldap.crt:
Using configuration from ca.config
Enter PEM pass phrase:
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName           :PRINTABLE:'CA'
stateOrProvinceName   :PRINTABLE:'Quebec'
localityName          :PRINTABLE:'Montreal'
organizationName      :PRINTABLE:'OpenNA, Inc.'
```

```
organizationalUnitName:PRINTABLE:'OpenNA, Inc. LDAP Server'
commonName             :PRINTABLE:'ldap.openna.com'
emailAddress           :IA5STRING:'noc@openna.com'
Certificate is to be certified until Mar 15 07:15:45 2002 GMT (365 days)
Sign the certificate? [y/n]: y


1 out of 1 certificate requests certified, commit? [y/n] y
Write out database with 1 new entries
Data Base Updated
CA verifying: ldap.crt <-> CA cert
ldap.crt: OK
```

This signs the CSR and results in a **ldap.crt** file.

Step 6
Now, we must place the certificates files (ldap.key and ldap.crt) to the appropriate directories and change their default permissions to be (0400/-r--------), owned by the user called 'ldap' for OpenLDAP to be able to find and use them when it starts its daemon.

- To place the certificates into the appropriate directory, use the following commands:

```
[root@deep ssl]# mv ldap.key private/
[root@deep ssl]# mv ldap.crt certs/
[root@deep ssl]# chmod 400 private/ldap.key
[root@deep ssl]# chmod 400 certs/ldap.crt
[root@deep ssl]# chown ldap.ldap private/ldap.key
[root@deep ssl]# chown ldap.ldap certs/ldap.crt
[root@deep ssl]# rm -f ldap.csr
```

First we move the ldap.key file to the private directory and the ldap.crt file to the certs directory. After that we change the permissions and ownership of both certificates to be only readable and owned by the OpenLDAP user called 'ldap' for security reasons. Finally we remove the ldap.csr file from our system since it is no longer needed.

Step 7
To allow TLS/SSL-enabled connections with OpenLDAP, we must specify two new options into the slapd.conf file. Text in bold is the parts of the lines that must be customized and adjusted to satisfy your needs.

- Edit the **slapd.conf** file (vi /etc/openldap/slapd.conf), and add the lines:

```
# See slapd.conf(5) for details on configuration options.
# This file should NOT be world readable.
include         /etc/openldap/schema/core.schema

# Do not enable referrals until AFTER you have a working directory
# service AND an understanding of referrals.
#referral       ldap://root.openldap.org

pidfile         /var/run/slapd.pid
argsfile        /var/run/slapd.args

# Load dynamic backend modules:
# modulepath    /usr/sbin/openldap
# moduleload    back_ldap.la
# moduleload    back_ldbm.la
# moduleload    back_passwd.la
```

```
# moduleload    back_shell.la

# Access Control List:
defaultaccess read
access to attr=userpassword
        by self write
        by dn="cn=Manager,dc=openna,dc=com" write
        by * compare

# Enable TLS/SSL connections with OpenLDAP:
TLSCertificateFile     /usr/share/ssl/certs/ldap.crt
TLSCertificateKeyFile  /usr/share/ssl/private/ldap.key

#######################################################################
# ldbm database definitions
#######################################################################

database        ldbm
readonly        off
suffix          "dc=openna,dc=com"
rootdn          "cn=Manager,dc=openna,dc=com"

# Cleartext passwords, especially for the rootdn, should
# be avoid. See slappasswd(8) and slapd.conf(5) for details.
# Use of strong authentication encouraged.
rootpw          secret

# The database directory MUST exist prior to running slapd AND
# should only be accessible by the slapd/tools. Mode 700 recommended.
directory       /var/lib/ldap

# Indices to maintain:
index uid       pres,eq
index cn,sn     pres,eq,sub
index objectClass eq
```

The `TLSCertificateFile` option specifies the file that contains the `slapd` server certificate, and the `TLSCertificateKeyFile` option specifies the file that contains the `slapd` server private key that matches the certificate stored in the `TLSCertificateFile` file.

Step 8
The `OpenLDAP` `TLS/SSL`-enabled connections run by default on port 636. To allow external traffic through this port (`636`), we must enable rules in our firewall script file for the **L**ightweight **D**irectory **A**ccess **P**rotocol (`LDAP`) server to accept external connections on the system.

Step 9
Finally, we must restart our `OpenLDAP` server for the changes to take effect.

- To restart `OpenLDAP` use the following command:

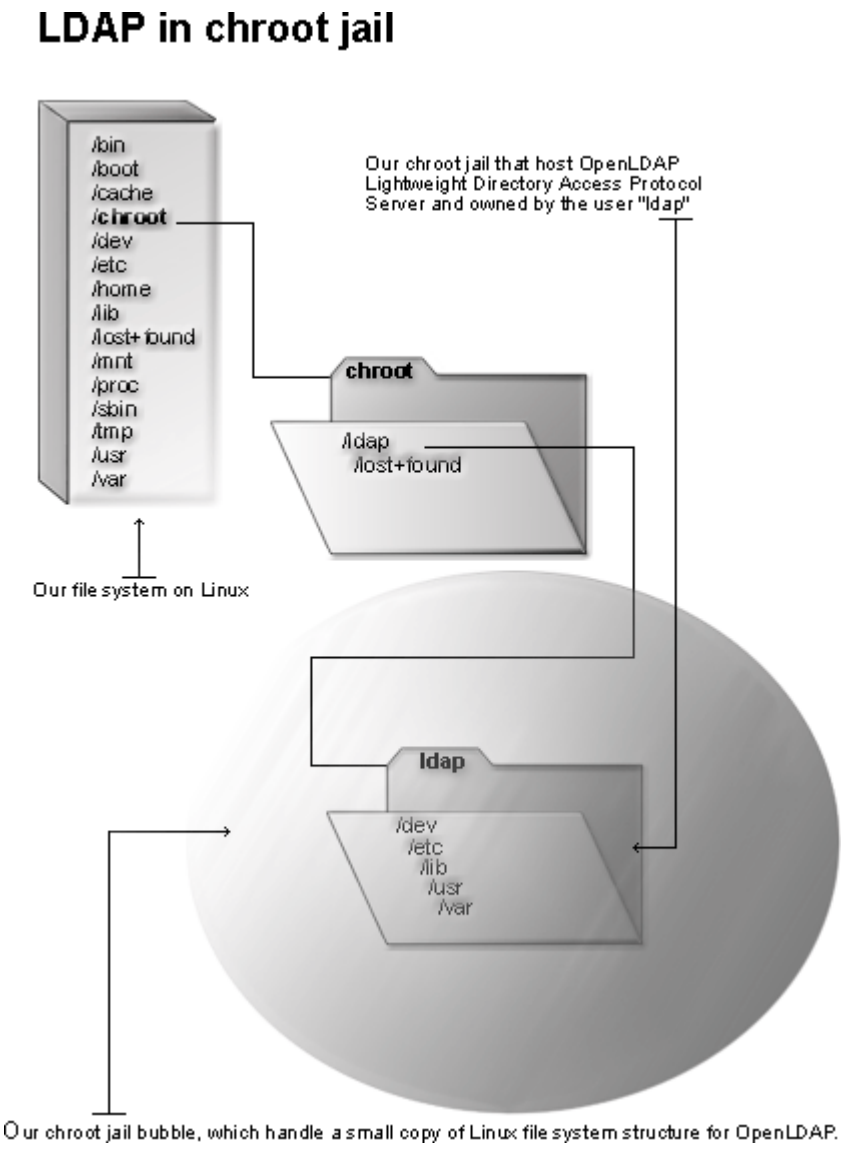
```
[root@deep /]# /etc/init.d/ldap restart
Stopping OpenLDAP:            [OK]
Starting OpenLDAP:            [OK]
```

Running `OpenLDAP` in a chroot jail

This part focuses on preventing `OpenLDAP` from being used as a point of break-in to the system hosting it. `OpenLDAP` by default runs **as a non-root user**, which will limit any damage to what can be done as a normal user with a local shell.

The main benefit of a chroot jail is that the jail will limit the portion of the file system the daemon can see to the root directory of the jail. Additionally, since the jail only needs to support `OpenLDAP`, the programs available into the jail can be extremely limited.

Most importantly, there is no need for setuid-root programs, which can be used to gain root access and break out of the jail. By running `OpenLDAP` in a chroot jail you can improve the security significantly in a UNIX environment.



Necessary steps to run OpenLDAP in a chroot jail:

What we're essentially doing is creating a skeleton root file system with enough components (directories, libraries, files, etc.) to allow UNIX to do a chroot when the OpenLDAP daemon starts.

Step 1

The first step to do for running OpenLDAP in a chroot jail will be to set up the chroot environment, and create the root directory of the jail. We've chosen /chroot/openldap for this purpose because we want to put this on its own separate file system to prevent file system attacks. Earlier in our Linux installation we created a special partition /chroot for this purpose.

```
[root@deep /]# /etc/init.d/ldap stop ← Only if OpenLDAP daemon already run.
Shutting down OpenLDAP:       [OK]

[root@deep /]# mkdir -p /chroot/openldap/dev
[root@deep /]# mkdir -p /chroot/openldap/lib
[root@deep /]# mkdir -p /chroot/openldap/etc
[root@deep /]# mkdir -p /chroot/openldap/usr/share
[root@deep /]# mkdir -p /chroot/openldap/usr/lib
[root@deep /]# mkdir -p /chroot/openldap/usr/sbin
[root@deep /]# mkdir -p /chroot/openldap/var/lib
[root@deep /]# mkdir -p /chroot/openldap/var/run
```

- For Red Hat Linux 7.3 users, you should create the following additional directory:
 `[root@deep /]# mkdir /chroot/openldap/lib/i686`

We need all of the above directories because, from the point of the chroot, we're sitting at "/" and anything above this directory is inaccessible.

Step 2

Next, it is important to move the main configuration directory, all configuration files, the database directory and the slapd binary program of the **L**ightweight **D**irectory **A**ccess **P**rotocol (LDAP) server to the chroot jail then create the special devices /dev/null and /dev/urandom which is/are absolutely required by the system to work properly.

```
[root@deep /]# mv /etc/openldap /chroot/openldap/etc/
[root@deep /]# mv /usr/share/openldap /chroot/openldap/usr/share/
[root@deep /]# mv /var/lib/ldap /chroot/openldap/var/lib/
[root@deep /]# mv /usr/sbin/slapd /chroot/openldap/usr/sbin/
[root@deep /]# mknod /chroot/openldap/dev/null c 1 3
[root@deep /]# chmod 666 /chroot/openldap/dev/null
[root@deep /]# mknod /chroot/openldap/dev/urandom c 1 9 ← Only for TLS/SSL.
```

NOTE: The /dev/urandom device is required only if you use TLS/SSL support with OpenLDAP.

Step 3
This step is required only if you have compiled OpenLDAP with TLS/SSL support. In this case, you must recreate a small copy of the /usr/share/ssl directory with certs and private directories which handles the private and public keys of OpenLDAP to the chroot jail environment.

- These procedures can be accomplished with the following commands:

```
[root@deep /]# mkdir -p /chroot/openldap/usr/share/ssl
[root@deep /]# mkdir -p /chroot/openldap/usr/share/ssl/certs
[root@deep /]# mkdir -p /chroot/openldap/usr/share/ssl/private
[root@deep /]# chown ldap.ldap /chroot/openldap/usr/share/ssl/certs/
[root@deep /]# chown ldap.ldap /chroot/openldap/usr/share/ssl/private/
[root@deep /]# cd /usr/share/ssl/
[root@deep ssl]# mv certs/ldap.crt /chroot/openldap/usr/share/ssl/certs/
[root@deep ssl]# mv private/ldap.key /chroot/openldap/usr/share/ssl/private/
```

Step 4
Now, we must find the shared library dependencies of slapd binary and install them into the chroot structure. Use the **ldd /chroot/openldap/usr/sbin/slapd** command to find out which libraries are needed. The output (depending on what you've compiled with OpenLDAP) will be something similar to:

- To find the shared library dependencies of slapd, execute the following command:

```
[root@deep /]# ldd /chroot/openldap/usr/sbin/slapd
libgdbm.so.2 => /usr/lib/libgdbm.so.2 (0x00129000)
libssl.so.2 => /lib/libssl.so.2 (0x00130000)
libcrypto.so.2 => /lib/libcrypto.so.2 (0x0015f000)
libcrypt.so.1 => /lib/libcrypt.so.1 (0x00233000)
libresolv.so.2 => /lib/libresolv.so.2 (0x00261000)
libdl.so.2 => /lib/libdl.so.2 (0x00273000)
libc.so.6 => /lib/libc.so.6 (0x00276000)
libgcc_s.so.1 => /lib/libgcc_s.so.1 (0x003ca000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x00110000)
```

What we can see here is the fact that depending of what programs have been compiled and included with OpenLDAP, the shared library dependencies may differ.

Step 5
Once the required libraries have been identified, copy them to the appropriate locations in the chroot jail. In our example these are the shared libraries identified above.

```
[root@deep /]# cp /usr/lib/libgdbm.so.2 /chroot/openldap/usr/lib/
[root@deep /]# cp /lib/libssl.so.2 /chroot/openldap/lib/
[root@deep /]# cp /lib/libcrypto.so.2 /chroot/openldap/lib/
[root@deep /]# cp /lib/libcrypt.so.1 /chroot/openldap/lib/
[root@deep /]# cp /lib/libresolv.so.2 /chroot/openldap/lib/
[root@deep /]# cp /lib/libdl.so.2 /chroot/openldap/lib/
[root@deep /]# cp /lib/libc.so.6 /chroot/openldap/lib/
[root@deep /]# cp /lib/libgcc_s.so.1 /chroot/openldap/lib/
[root@deep /]# cp /lib/ld-linux.so.2 /chroot/openldap/lib/
```

You'll also need the following extra libraries for some network functions, like resolving etc.:

```
[root@deep /]# cp /lib/libnss_compat* /chroot/openldap/lib/
[root@deep /]# cp /lib/libnss_dns* /chroot/openldap/lib/
[root@deep /]# cp /lib/libnss_files* /chroot/openldap/lib/
[root@deep /]# strip -R .comment /chroot/openldap/lib/*
[root@deep /]# strip -R .comment /chroot/openldap/usr/lib/*
```

- For Red Hat Linux 7.3 users, you should copy the following additional library:
 `[root@deep /]# cp /lib/i686/libc.so.6 /chroot/openldap/lib/i686/`

NOTE: The "`strip -R .comment`" commands will remove all the sections named "`.comment`" from the libraries files under the `/usr/lib` and `/lib` directory of the chroot jail and will make them smaller in size to increase performance.

Step 6
Now we need to copy the **`passwd`** and **`group`** files inside the `/chroot/openldap/etc` directory. Next, we'll remove all entries except for the user that `slapd` runs as in both files.

```
[root@deep /]# cp /etc/passwd /chroot/openldap/etc/
[root@deep /]# cp /etc/group /chroot/openldap/etc/
```

- Edit the **`passwd`** file under the chroot jail (`vi /chroot/openldap/etc/passwd`) and delete all entries except for the user `slapd` runs as (in our configuration, it's "`ldap`"):

  ```
  ldap:x:55:55:OpenLDAP Server:/var/lib/ldap:/bin/false
  ```

- Edit the **`group`** file under the chroot jail (`vi /chroot/openldap/etc/group`) and delete all entries except the group `slapd` runs as (in our configuration it's "`ldap`"):

  ```
  ldap:x:55:
  ```

Step 7
You will also need **`resolv.conf`**, **`nsswitch.conf`**, **`localtime`** and **`hosts`** files in your chroot jail structure.

```
[root@deep /]# cp /etc/resolv.conf /chroot/openldap/etc/
[root@deep /]# cp /etc/nsswitch.conf /chroot/openldap/etc/
[root@deep /]# cp /etc/localtime /chroot/openldap/etc/
[root@deep /]# cp /etc/hosts /chroot/openldap/etc/
```

Step 8
Now we must set some of the files in the chroot jail directory immutable for better security.

- These procedures can be accomplished with the following commands:

```
[root@deep /]# cd /chroot/openldap/etc/
[root@deep etc]# chattr +i passwd
[root@deep etc]# chattr +i group
[root@deep etc]# chattr +i resolv.conf
[root@deep etc]# chattr +i hosts
[root@deep etc]# chattr +i nsswitch.conf
```

WARNING: Don't forget to remove the immutable bit on these files if you have to modify them later using the command "`chattr -i`".

Step 9
The default `ldap` initialization script of `OpenLDAP` starts the daemon "`slapd`" and/or "`slurpd`" outside the chroot jail. We must change it now to start `slapd` and/or `slurpd` from the chroot jail environment.

Since there were so many lines to modify from the original initialization script file of `OpenLDAP` to make it start in the jail environment, I decided to make a new initialization file as shown below. Lines in bold are the ones that are different from the original script file. In this way you'll be able to see how I've changed it.

- Edit the **`ldap`** script file (`vi /etc/init.d/ldap`) and add/change the following lines:

```
#!/bin/bash

# This shell script takes care of starting and stopping OpenLDAP.
#
# chkconfig: 345 39 61
# description: LDAP is a Lightweight Directory Access Protocol, used \
#              for implementing the industry standard directory services.
#
# processname: slapd
# config: /chroot/openldap/etc/openldap/slapd.conf
# pidfile: /var/run/slapd.pid

# Source function library.
. /etc/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Source for additional options if we have them.
if [ -f /etc/sysconfig/ldap ] ; then
        . /etc/sysconfig/ldap
fi

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

# Some definition for easy maintenance.
ROOTDIR=/chroot/openldap
```

```
# If OpenLDAP is not available stop now.
[ -f $ROOTDIR/usr/sbin/slapd ] || exit 0
[ -f $ROOTDIR/etc/openldap/slapd.conf ] || exit 0

# Path to the OpenLDAP binaries.
slapd=$ROOTDIR/usr/sbin/slapd
slurpd=$ROOTDIR/usr/sbin/slurpd

RETVAL=0
prog="OpenLDAP"

start() {
        echo -n $"Starting $prog: "
        if grep -q ^TLS $ROOTDIR/etc/openldap/slapd.conf ; then
        daemon $slapd -u ldap -r $ROOTDIR -h '"ldap:/// ldaps:///"'
            RETVAL=$?
        else
        daemon $slapd -u ldap -r $ROOTDIR
            RETVAL=$?
        fi
        echo
        if [ $RETVAL -eq 0 ]; then
            if grep -q "^replogfile" $ROOTDIR/etc/openldap/slapd.conf;
then
                echo -n $"Starting $prog: "
                daemon $slurpd -r $ROOTDIR
                RETVAL=$?
                echo
            fi
        fi
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/ldap
        return $RETVAL
}

stop() {
        echo -n $"Shutting down $prog: "
        killproc $slapd
        RETVAL=$?
        echo
        if [ $RETVAL -eq 0 ]; then
            if grep -q "^replogfile" $ROOTDIR/etc/openldap/slapd.conf;
then
                echo -n $"Shutting down $prog: "
                killproc $slurpd
                RETVAL=$?
                echo
            fi
        fi
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/ldap
/var/run/slapd.args
        return $RETVAL
}

# See how we were called.
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  status)
```

```
        status $slapd
        if grep -q "^replogfile" $ROOTDIR/etc/openldap/slapd.conf ; then
           status $slurpd
        fi
        ;;
  restart)
        stop
        start
        RETVAL=$?
        ;;
  condrestart)
        if [ -f /var/lock/subsys/ldap ] ; then
            stop
            start
            RETVAL=$?
        fi
        ;;
  *)
        echo $"Usage: $0 {start|stop|status|restart|condrestart}"
        exit 1
esac
exit $RETVAL
```

Step 10
Finally, we must test the new chrooted jail configuration of our **L**ightweight **D**irectory **A**ccess **P**rotocol (LDAP) server.

- Start the new chrooted jail OpenLDAP with the following command:
  ```
  [root@deep /]# /etc/init.d/ldap start
  Starting OpenLDAP:           [OK]
  ```

- If you don't get any errors, do a **ps ax | grep slapd** and see if we're running:
  ```
  [root@deep /]# ps ax | grep slapd
  10022 ?        S      0:00 /chroot/openldap/usr/sbin/slapd -u ldap -r
  /chroot/openldap
  ```

If so, lets check to make sure it's chrooted by picking out its process number and doing **ls -la /proc/that_process_number/root/**.

```
[root@deep /]# ls -la /proc/10022/root/
```

If you see something like:

```
drwxr-xr-x    7 root     root         4096 Jun  3 00:07 ./
drwxr-xr-x    5 root     root         4096 Jun  3 00:07 ../
drwxr-xr-x    2 root     root         4096 Jun  3 00:07 dev/
drwxr-xr-x    3 root     root         4096 Jun  3 00:07 etc/
drwxr-xr-x    2 root     root         4096 Jun  3 00:07 lib/
drwxr-xr-x    5 root     root         4096 Jun  3 00:07 usr/
drwxr-xr-x    4 root     root         4096 Jun  3 00:07 var/
```

Congratulations!

Securing OpenLDAP

This section deals specifically with actions we can take to improve and tighten security under OpenLDAP **L**ightweight **D**irectory **A**ccess **P**rotocol (LDAP) server. The interesting point here is that we refer to the features available within the base installed program and not to any additional software.

Using an encrypted root password:

With a default installation of OpenLDAP, clear text passwords for the rootdn are used. Use of strong authentication is encouraged using the slappasswd command utility of the directory server.

Below, I show you how to use an encrypted root password, which is a much better idea than leaving a plain text root password in the slapd.conf file.

Step 1
Our first action will be to use the slappasswd tool of OpenLDAP to generate hashed passwords. The utility will prompt you to enter (twice) the user password that you want it to generate in an encrypted form. The schemes that we must generate is a so called (CRYPT) and we specify it with the "-h" option during hashed password generation.

```
[root@deep /]# /usr/sbin/slappasswd -h {CRYPT}
New password:
Re-enter new password:
{CRYPT}0f.piDw01Vi7w
```

Here the generated "{CRYPT}0f.piDw01Vi7w" line is the one that we must copy into the /etc/openldap/slapd.conf file to replace the old clear text password for the rootdn.

Step 2
Once we get the generated hashed password line for our rootdn, we must edit the slapd.conf file and add it to the rootpw line.

- Edit the **slapd.conf** file (vi /etc/openldap/slapd.conf) and change the line:

  ```
  rootpw          secret
  ```

 To read:

  ```
  rootpw          {CRYPT}0f.piDw01Vi7w
  ```

NOTE: Use of hashed passwords does not protect passwords during protocol transfer. TLS or other eavesdropping protections should be in place before using LDAP's simple bind. The hashed password values should be protected as if they were clear text passwords.

Immunize important configuration files:

The immutable bit can be used to prevent one from accidentally deleting or overwriting a file that must be protected. It also prevents someone from creating a symbolic link to this file. Once your `slapd.conf` file has been configured, it's a good idea to immunize it with command like:

```
[root@deep /]# chattr +i /etc/openldap/slapd.conf
```

or:

```
[root@deep /]# chattr +i /chroot/openldap/etc/openldap/slapd.conf
```

if you are running `OpenLDAP` in chroot jail environment.

Optimizing `OpenLDAP`

This section deals specifically with actions we can make to improve and tighten performance of `OpenLDAP` **L**ightweight **D**irectory **A**ccess **P**rotocol (`LDAP`) server. Note that we refer to the features available within the base installed program.

Get some fast `SCSI` hard disk:

One of the most important parts of optimizing an `OpenLDAP` server, as well as for the majority of all `SQL` database servers, is the speed of your hard disk, the faster it is, and the faster your database will run. Consider a `SCSI` disk with low seek times, like `4.2ms`, this can make all the difference, much greater performance can also be made using `RAID` technology.

Skip the updating of the last access time:

As you're supposed to know now, the **`noatime`** attribute of Linux eliminates the need by the system to make writes to the file system for files. Mounting the file system where your `OpenLDAP` **L**ightweight **D**irectory **A**ccess **P**rotocol (`LDAP`) server lives with the `noatime` attribute will avoid some disk seeks and will improve the performance of you directory server.

If you want to mount the file system of the `OpenLDAP` **L**ightweight **D**irectory **A**ccess **P**rotocol (`LDAP`) server with the `noatime` attribute, it's important to create and install its databases on this partition. In our example, we have created this partition early in chapter 2 of this book (Linux Installation) and this partition is located on `/var/lib`.

Step 1

To mount the file system of `OpenLDAP` **L**ightweight **D**irectory **A**ccess **P**rotocol (`LDAP`) server with the `noatime` option, you must edit the **`fstab`** file (`vi /etc/fstab`) and add to the line that refers to the `/var/lib` file system the `noatime` option as shown below:

- Edit the **`fstab`** file (`vi /etc/fstab`), and change the line:

  ```
  LABEL=/var/lib    /var/lib        ext3    defaults        1 2
  ```

 To read:

  ```
  LABEL=/var/lib    /var/lib        ext3    defaults,noatime        1 2
  ```

> **NOTE**: The line related to `/var/lib` in your `/etc/fstab` file could be different from the one above, this is just an example. Also, if you are running `OpenLDAP` in chroot jail environment, the file system to mount with the `noatime` option will be `/chroot` and not `/var/lib`.

Step 2
Once you have made the necessary adjustments to the /etc/fstab file, it is time to inform the system about the modification.

- This can be done with the following command:
```
[root@deep /]# mount /var/lib -oremount
```

Each file system that has been modified must be remounted with the command as shown above. In our example we have modified the /var/lib file system.

Step 3
After your file system has been remounted, it is important to verify that if the modifications made in the fstab file have been correctly applied.

- You can verify if the modifications have been correctly applied with the command:
```
[root@deep /]# cat /proc/mounts
/dev/root       /              ext3 rw 0 0
/proc           /proc          proc rw 0 0
/dev/sda1       /boot          ext3 rw 0 0
/dev/sda9       /chroot        ext3 rw 0 0
/dev/sda8       /home          ext3 rw 0 0
/dev/sda13      /tmp           ext3 rw 0 0
/dev/sda7       /usr           ext3 rw 0 0
/dev/sda11      /var           ext3 rw 0 0
/dev/sda12      /var/lib       ext3 rw,noatime 0 0
none            /dev/pts       devpts rw 0 0
```

This command will show you all file systems on your server with the parameters applied to them. If you see something like:

```
/dev/sda12    /var/lib      ext3   rw,noatime 0 0
```
Congratulations!

> **NOTE:** Look under the chapter related to the Linux Kernel in this book for more information about the noatime attribute and other tunable parameters.

OpenLDAP Administrative Tools
The commands listed below are some that we use often, but many more exist. Check the manual pages of OpenLDAP and documentation for more information.

Creating an LDMB backend database:
There are two methods of creating a database for LDAP, the first is off-line, with the slapadd command utility, and the other is on-line, with the ldapadd command utility.

Usually you use the off-line method when you have many thousands of entries to insert into your database and the on-line method when you have only a small number of entries to put into your database. It is also important to note that the off-line method requires that the slapd daemon is NOT running and the on-line method requires that the slapd daemon of OpenLDAP is running.

slapadd

When you install OpenLDAP for the first time and have a large number of entries to put in your backend database, it's always a good idea to put all this information into a text file and add them to your backend database with the slapadd command utility.

This command is used to create the LDMB backend database off-line. To do it, the first thing will be to create an LDIF (**L**DAP **D**ata **I**nterchange **F**ormat) input file containing a text representation of your entries. So to summarize, the slapadd tool of OpenLDAP converts an LDIF file into an LDBM back-end database.

Step 1

The text file named "datafiles" below can be used as an example file (of course, your real LDIF input file will handle much more information than this example). A blank line indicates that the entry is finished and that another entry is about to begin.

- Create the **datafiles** file (touch /tmp/datafiles) and add as an example in this file the following lines:

```
# Organization's Entry
dn: dc=openna,dc=com
dc: openna
objectclass: dcObject
objectclass: organization
o: OpenNA.com, Inc.
#
# Gerhard's Entry
dn: cn=Gerhard Mourani,dc=openna,dc=com
cn: Gerhard Mourani
sn: Mourani
objectclass: organizationalRole
objectclass: organizationalPerson
#
# Ted's Entry
dn: cn=Ted Nakad,dc=openna,dc=com
cn: Ted Nakad
sn: Nakad
description: Agent & Sales Manager
objectclass: organizationalRole
objectclass: organizationalPerson
```

The above entries give you some very basic examples about how to convert your information into LDIF files before adding them to your new backend directory. Consult the OpenLDAP documentation and especially books for more information.

WARNING: Before adding any objects to the database, you have to add an entry for your organization, first. This is done with the following in the above example.

```
dn: dc=openna,dc=com
dc: openna
objectclass: dcObject
objectclass: organization
o: OpenNA.com Inc.
```

Please note that these entries only have to be entered once to create your organization, after that all you have to do is to add additional information as we do for Gerhard's and Ted's.

Step 2
Once the LDIF input file containing our entries has been created, we must insert them into the **L**ightweight **D**irectory **A**ccess **P**rotocol (LDAP) server.

- To insert the LDIF input file and create the database off-line, use the commands:
```
[root@deep /]# cd /tmp/
[root@deep tmp]# slapadd -l datafiles
```

The "-l" option specifies the location of the LDIF input file (datafiles) containing the entries in text form to add.

WARNING: The above command cannot work if OpenLDAP is started in the chroot jail environment. The slapd daemon of OpenLDAP is not started in this mode. Be sure to replace all the required information with the appropriate domain components of your domain name.

ldapadd

If the entries in your directory server are already created or if you have only a small amount of information to insert into your backend database, you'd probably prefer to use the ldapadd command utility to do your job on-line. The ldapadd utility is used to add entries to your directory while the LDAP server is running and expects input in LDIF (**L**DAP **D**ata **I**nterchange **F**ormat) form.

Step 1
For example, to add the "Europe Mourani" entry using the ldapadd tool, you could create a file called "entries" with input in LDIF form into your /tmp directory.

- Create the **entries** file (touch /tmp/entries) and add to it, as an example, the following contents:

```
# Organization's Specifications
dn: dc=openna,dc=com
dc: openna
objectclass: dcObject
objectclass: organization
o: OpenNA.com Inc.
#
# Europe's Entry
dn: cn=Europe Mourani,dc=openna,dc=com
cn: Europe Mourani
sn: Mourani
description: Marketing Representatif
objectclass: organizationalRole
objectclass: organizationalPerson
```

Step 2
Once the **entries** file has been created, we must add its contents into the OpenLDAP server.

- To actually create the entry on-line in the backend database, use the commands:

```
[root@deep /]# cd /tmp/
[root@deep tmp]# ldapadd -f entries -D "cn=Manager, dc=openna, dc=com" -W
Enter LDAP Password :
adding new entry "dc=openna,dc=com"

adding new entry "cn=Europe Mourani,dc=openna,dc=com"
```

The above command assumes that you have set your rootdn to "cn=Manager,dc=openna, dc=com" and rootpw to an encrypted root password. You will be prompted to enter the encrypted root password.

NOTE: The slapd daemon of OpenLDAP is started in this creation mode. Be sure to replace all the required information with the appropriate domain components of your domain name.

ldapmodify
Contrary to relational databases, where data is constantly changed, the directory server contains information that is rarely modified once inserted. But, sometimes you need to modify information, and the ldapmodify tool will help you. The ldapmodify command allows you to modify entries on the backend directory server.

Step 1
Assuming that we want to replace the contents of the "Europe Mourani" entry's description attribute with the new value "Marketing Representative", the following will achieve it.

- Create the **lnew** file (touch /tmp/lnew) and add the following:

```
dn: cn=Europe Mourani,dc=openna,dc=com
changetype: modify
replace: description
description: Marketing Representative
```

Step 2
Once the lnew file has been created, we must replace the entry in the OpenLDAP directory server with the one contained in this file (lnew).

- To modify the contents of backend database, use the following commands:

```
[root@deep /]# cd /tmp/
[root@deep tmp]# ldapmodify -f lnew -D 'cn=Manager, dc=openna, dc=com' -W
Enter LDAP Password:
modifying entry "cn=Europe Mourani,dc=openna,dc=com"
```

OpenLDAP Users Tools

The commands listed below are some that we use often, but many more exist. Check the manual pages of OpenLDAP and other documentation for more information.

ldapsearch

The ldapsearch utility searches through the backend database of the LDAP directory for the information/entries you have requested.

- To search on LDAP directory for entries, use the following command:

```
[root@deep /]# ldapsearch -b 'dc=openna, dc=com' 'cn=europe*'
version: 2

#
# filter: cn=europe*
# requesting: ALL
#

# Europe Mourani,dc=openna,dc=com
dn: cn=Europe Mourani,dc=openna,dc=com
cn: Europe Mourani
sn: Mourani
objectClass: organizationalRole
objectClass: person
description: Marketing Representative

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

This command will retrieve all entries and values for the name **europe** and will print the results to standard output in your terminal.

Further documentation

For more details, there are several manual pages for OpenLDAP that you can read; below I show you just the most important ones:

```
$ man ldapd (8)                 - LDAP X.500 Protocol Daemon.
$ man ldapdelete (1)            - LDAP delete entry tool.
$ man ldapfilter.conf (5)       - Configuration file for LDAP get filter routines.
$ man ldapfriendly (5)          - Data file for LDAP friendly routines.
$ man ldapmodify, ldapadd (1)   - LDAP modify entry and ldap add entry tools.
$ man ldapmodrdn (1)            - LDAP modify entry RDN tool.
$ man ldappasswd (1)            - Change the password of an LDAP entry.
$ man ldapsearch (1)            - LDAP search tool.
$ man ldapsearchprefs.conf (5)  - Configuration file for LDAP search preference routines.
$ man ldaptemplates.conf (5)    - Configuration file for LDAP display template routines.
$ man ldif (5)                  - LDAP Data Interchange Format.
$ man slapd (8)                 - Stand-alone LDAP Daemon.
$ man slapd.conf (5)            - Configuration file for slapd, the stand-alone LDAP daemon.
$ man slurpd (8)                - Standalone LDAP Update Replication Daemon.
$ man ud (1)                    - Interactive LDAP Directory Server query program.
```



Part XIII

File Transfer Protocol

CHAPTER 41

ProFTPD

IN THIS CHAPTER

1. Compiling - Optimizing & Installing `ProFTPD`
2. Configuring `ProFTPD`
3. Creating an account for `FTP` client to connect to the `FTP` server
4. Setup an anonymous `FTP` server
5. Allow anonymous users to upload to the `FTP` server
6. Running `ProFTPD` with `SSL` support
7. Securing `ProFTPD`
8. `ProFTPD` Administrative Tools

Linux ProFTPD

Abstract

Despite its age, using the **F**ile **T**ransfer **P**rotocol (FTP) is one of the most popular ways to transfer files from machine to machine across a network. Clients and servers have been written for each of the popular platforms on the market, thereby making FTP the most convenient way to perform file transfers between different platforms.

Many different ways exist to configure your FTP servers. One is as a local user-only site, which is the default configuration for an FTP server; a local users FTP server allows users from any kind of operating system having FTP client software to connect via the FTP server and access their files.

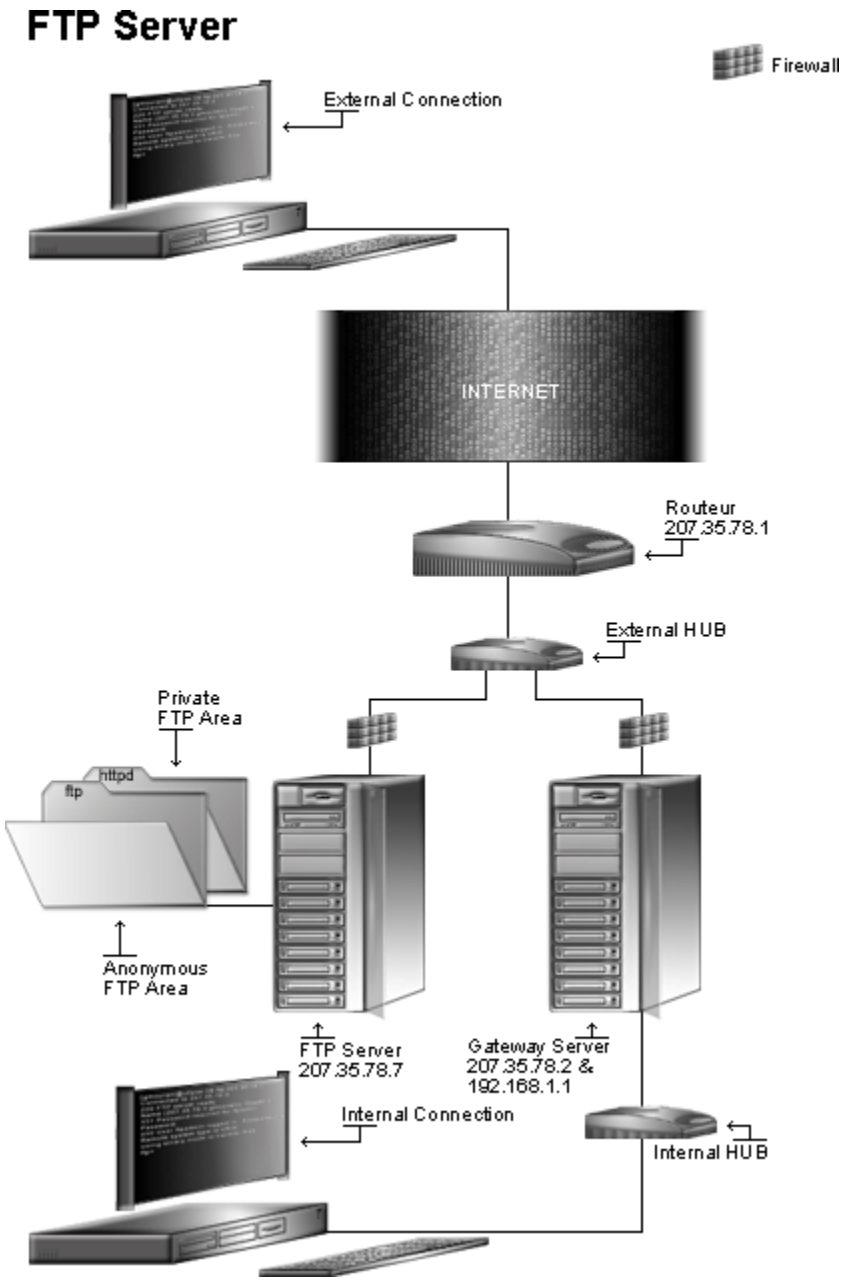
Other kinds exist, like the anonymous FTP server. An anonymous FTP server allows anyone on the network to connect to it and transfer files without having an account. Due to the potential security risk involved with this setup, precautions should be taken to allow access only to certain directories on the system.

The configuration we will cover in this chapter is an FTP server that allows FTP to semi-secure areas of a UNIX file system (chroot'd FTP access). This configuration allows users to have access to the FTP server directories without allowing them to get into higher levels. This is the most secure setup for an FTP server and it is a useful way for remote clients to maintain their Web accounts.

ProFTPD is a secure and highly configurable FTP server for Linux. It has been designed to be much like Apache in concept, taking many of the ideas (configuration format, modular design, etc) from it. If you are comfortable with Apache web server configuration, you'll find that the ProFTPD configuration file is easy to understand and setup.

As secure by design as possible, it offers the feature set required for the more sophisticated FTP sites today. ProFTPD is the perfect secure FTP server for sites that offer web hosting to their customers; it is also the perfect FTP server for virtual web hosting. ProFTPD does not sacrifice security or ease of use.

In this FTP section of the book, we'll begin our discussion about FTP servers with ProFTPD and will propose you an alternative for those who only need to transfer files from one place to another without the need of complete FTP functionality. Some of us do not provide any kind of web hosting or customers services via an FTP server and just need to have a secure and fast FTP server to transfer files from machine to machine. For these people, we will explain how to compile, install, and configure vsFTPd in the next chapter.



In the above schema, you can see that client machines go through the `FTP` server from different ways. They can come from the internal network or from the external network. Both client `FTP` connections can be made on the local user `FTP` area or the anonymous `FTP` area.

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest ProFTPD version number is 1.2.5

The procedures given in this chapter are likely to work on all Linux platforms, but we have only tested it on OpenNA Linux and Red Hat Linux.

Packages

The following is based on information as listed by ProFTPD as of 2002/06/09. Please regularly check at http://proftpd.linux.co.uk/ for the latest status. We chose to install the required component from source file because it provides the facility to fine tune the installation.

Source code is available from:
ProFTPD Homepage: http://proftpd.linux.co.uk/
ProFTPD FTP Site: 216.10.40.219
You must be sure to download: proftpd-1.2.5.tar.gz

Prerequisites

ProFTPD requires that the software below is already installed on your system to be able to compile successfully. If this is not the case, you must install it. Please make sure you have this program installed on your machine before you proceed with this chapter.

- ✓ OpenSSL is required to run ProFTPD with SSL support on your system.

> **NOTE:** For more information about OpenSSL software, see its related chapter in this book.

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all installed files into the system in the eventuality of an updated in the future. To solve the problem, it is a good idea to make a list of files on the system before you install ProFTPD, and one afterwards, and then compares them using the **diff** utility to find out what files are placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep root]# find /* > ProFTPD1
  ```
- And the following one after you install the software:
  ```
  [root@deep root]# find /* > ProFTPD2
  ```
- Then use the following command to get a list of what changed:
  ```
  [root@deep root]# diff ProFTPD1 ProFTPD2 > ProFTPD-Installed
  ```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. Related to our example above, we use the /root directory of the system to stock all generated list files.

Compiling - Optimizing & Installing ProFTPD

Below are the required steps that you must make to configure, compile and optimize the ProFTPD software before installing it into your Linux system. First off, we install the program as user 'root' so as to avoid authorization problems.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- This can be done with the following commands:

```
[root@deep /]# cp proftpd-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf proftpd-version.tar.gz
[root@deep tmp]# cd proftpd-version
```

Step 2
In order to check that the version of ProFTPD, which you are, going to install, is an original and unmodified one, please check the supplied signature with the PGP key of ProFTPD available on the ProFTPD website.

To get a PGP key copy of ProFTPD, please point your browser to the following URL: http://proftpd.linux.co.uk/pgp.html. For more information about how to use this key for verification, see the GnuPG chapter in this book.

Step 3
ProFTPD cannot run as super-user root; for this reason we must create a special user with no shell privileges on the system for running ProFTPD daemon.

- To create this special ProFTPD user on OpenNA Linux, use the following command:

```
[root@deep tmp]# groupadd -g 24 ftp > /dev/null 2>&1 || :
[root@deep tmp]# useradd -c "FTP Server" -d /home/ftp -g 24 -s
/bin/false -u 24 ftp > /dev/null 2>&1 || :
```

- To create this special ProFTPD user on Red Hat Linux, use the following command:

```
[root@deep tmp]# groupadd -g 24 ftp > /dev/null 2>&1 || :
[root@deep tmp]# useradd -u 24 -g 24 -s /bin/false -M -r -d /home/ftp
ftp > /dev/null 2>&1 || :
```

The above command will create a null account, with no password, no valid shell, no files owned-nothing but a UID and a GID for the program. Remember that ProFTPD daemon does not need to have a shell account on the server.

Step 4
Now, edit the **shells** file (`vi /etc/shells`) and add a non-existent shell name "`/bin/false`", which is the one we used in the `useradd` command above.

```
[root@deep tmp]# vi /etc/shells
/bin/bash2
/bin/bash
/bin/sh
/bin/false ← This is our added no-existent shell
```

Step 5
After that, move into the newly created `ProFTPD` directory and perform the following step before compiling and optimizing it. The modification we make to the `ProFTPD` source file below is necessary to improve the default internal buffer size used for `FTP` data transfers and other miscellaneous tasks with `ProFTPD`.

Be aware that this modification will not work on all Linux systems. To be able to use this hack on your Linux server, you should be sure that maximum number of open file descriptors can be set at least to 8192. If this is not possible, then skip this step.

- To verify if the maximum number of open file descriptors can be set at least to 8192 on your Linux system, use the following command:
 `[root@deep proftpd-1.2.5]# ulimit -n 8192`

NOTE: If the above command returns an error message like: "`bash: ulimit: cannot modify open files limit: Operation not permitted`", then your Linux server cannot support this hack and you should not do it on your system. OpenNA Linux is known to support this hack.

Step 6
The file that we must modify to improve the default internal buffer size is called **options.h** located under the `include/` directory of `ProFTPD`. In this file, we will change the default setting.

- Edit the **options.h** file (`vi +73 include/options.h`) and change the following value:

  ```
  #define TUNABLE_BUFFER_SIZE    1024
  ```

 To read:

  ```
  #define TUNABLE_BUFFER_SIZE    8192
  ```

Patching ProFTPD to compile with SSL support:

There is an external patch available for ProFTPD that allows us to compile ProFTPD with SSL support. If you are interested in compiling ProFTPD to support SSL encryption of usernames and passwords on the FTP server, then I recommend you follow these steps. If you don't want to compile ProFTPD with SSL support, you can simply skip these steps and go directly to next section where we will compile the software for our system.

Also, it's important to note that SSL support with ProFTPD is required ONLY if you want to setup your FTP server for so called local users FTP connections, you really don't need to compile ProFTPD with SSL support if you intended to run your FTP server for anonymous connections.

Finally, not all FTP client software provides SSL support with FTP servers; you have to be sure that the FTP client that you or your customers use to connect to the FTP server can support SSL connections.

Step 1
First off, we have to retrieve the SSL patch which is available on the Internet. This patch can be downloaded from the following location: ftp://ftp.runestig.com/pub/proftpd-tls/

Step 2
Once you have a copy of this patch, you should move it under the /var/tmp directory and patch your ProFTPD source files.

- This can be done with the following commands:

```
[root@deep /]# mv proftpd-1.2.5-tls-20020617.patch.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# gunzip proftpd-1.2.5-tls-20020617.patch.gz
[root@deep tmp]# patch -p0 < proftpd-1.2.5-tls-20020617.patch
```

NOTE: It's important to note that the version number of the SSL patch that you have to download from the Internet must match the version number of the ProFTPD software you intend to install. For example, if the version number of ProFTPD is 1.2.5, you should download the newer SSL patch that matches this number.

Compiling ProFTPD:

Once all the required modifications have been made to ProFTPD as shown above (and only if required), it is time compile and optimize ProFTPD for our system.

Step 1
Compile and optimize ProFTPD with the following compilation lines.

- To compile and optimize ProFTPD use the following compilation lines:

```
CFLAGS="-O2 -march=i686 -funroll-loops"; export CFLAGS
./configure \
--prefix=/usr \
--sysconfdir=/etc \
--localstatedir=/var/run \
--mandir=/usr/share/man \
--enable-pam \
--with-openssl-dir=/usr/share/ssl \
--with-modules=mod_linuxprivs:mod_readme:mod_quota
```

This tells ProFTPD to set itself up for this particular configuration setup with:

- Enable PAM support.
- Specify location of the OpenSSL directory.
- Add additional mod_linuxprivs module to proftpd.
- Add additional mod_readme module to proftpd.
- Add additional mod_quota module to proftpd.

ProFTPD, like the Apache web server, uses the concept of modules to add additional features to the FTP server. The option "–with-modules" allows us to compile the FTP server with any available modules of our choice. In the above compilation, we enable modules support for the following features with ProFTPD:

1) mod_linuxprivs: This security module allows ProFTPD to successfully drop all of the capabilities that gives 'root' privileges and provide fine-grained control over what operations are allowed on the FTP server. It is a very good security module to compile with ProFTPD.

2) mod_readme: This module allows ProFTPD to display "readme" files on the FTP server (if required).

3) mod_quota: This module is really useful when you want to provide directory tree based disk quotas via the FTP server. With mod_quota, you don't need anymore to control disk quotas on users FTP directories with external tool like quota.

To get the list of all available modules that you may use and compile with ProFTPD, please read the README.modules file into the source directory of ProFTPD.

NOTE: The option "--with-openssl-dir=/usr/share/ssl" is required ONLY if you have patched ProFTPD with the SSL patch and want to compile ProFTPD with SSL support. If you don't want to run the FTP server with SSL support, you have to remove this option from the list.

Step 2
At this stage of our work the program is ready to be built and installed. We build ProFTPD with the 'make' command and produce a list of files on the system before we install the software, and one afterwards, then compare them using the **diff** utility to find out what files are placed where and finally install ProFTPD.

```
[root@deep proftpd-1.2.5]# make
[root@deep proftpd-1.2.5]# cd
[root@deep root]# find /* > ProFTPD1
[root@deep root]# cd /var/tmp/proftpd-1.2.5/
[root@deep proftpd-1.2.5]# make install
[root@deep proftpd-1.2.5]# strip /usr/sbin/proftpd
[root@deep proftpd-1.2.5]# strip /usr/sbin/ftpshut
[root@deep proftpd-1.2.5]# strip /usr/bin/ftpcount
[root@deep proftpd-1.2.5]# strip /usr/bin/ftpwho
[root@deep proftpd-1.2.5]# cd
[root@deep root]# find /* > ProFTPD2
[root@deep root]# diff ProFTPD1 ProFTPD2 > ProFTPD-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

Step 3
Once the compilation, optimization and installation of the software have been finished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete ProFTPD and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf proftpd-version/
[root@deep tmp]# rm -f proftpd-version.tar.gz
```

Configuring ProFTPD

After ProFTPD has been built and installed successfully in your system, your next step is to configure and customize its configuration files to fit your needs.

- ✓ /etc/proftpd.conf (The ProFTPD Configuration File)
- ✓ /etc/sysconfig/proftpd (The ProFTPD System Configuration File)
- ✓ /etc/pam.d/ftp (The ProFTPD PAM Support Configuration File)
- ✓ /etc/ftpusers (The ProFTPD Access Configuration File)
- ✓ /etc/rc.d/init.d/proftpd (The ProFTPD Initialization File)

/etc/proftpd.conf: The ProFTPD Configuration File

The /etc/proftpd.conf file is the main configuration file for ProFTPD. It is in this configuration file that ProFTPD gets all of its information and the way it should run on your system. We can configure the proftpd.conf file to run ProFTPD as an anonymous FTP server, or as a local users FTP server for web hosting, etc.

Different configurations exist, and we will show you later how to configure it in the most secure way when running as a local users FTP server and also as an anonymous FTP server. We start our configuration by showing you how to configure it to run as a local users FTP server, which is the most complex configuration.

ProFTPD uses the concept of configuration directives with contexts. At present, ProFTPD has seven different configuration contexts: <General>, <Global>, <Limit>, <Anonymous>, <Directory>, <VirtualHost>, and .ftpaccess files.

In our configuration example, we will use some of these contexts to achieve our aims. We will separate and interpret each context as separate sections.

- ✓ One section for the general configuration of the FTP server "General Server Context";
- ✓ One section for the global configuration of the FTP server "Global Server Context";
- ✓ One section for the limit configuration of the FTP server "Limit Server Context";
- ✓ One section for the anonymous configuration of the FTP server "Anonymous Server Context".

NOTE: It's important to note that other contexts like `<Directory>` are also used in the `ProFTPD` configuration as shown during the configuration of the `proftpd.conf` file. Depending of your requirements and needs, you'll use different contexts in different places in the configuration file.

- Edit the **`proftpd.conf`** file (`vi /etc/proftpd.conf`) and set your requirements. Below is what we recommend you use for local users `FTP` access:

```
# General Server Context.
ServerName                      "OpenNA Linux"
ServerType                      standalone
DefaultServer                   on
Port                            21
tcpBackLog                      10
MaxInstances                    30
CommandBufferSize               50
UseReverseDNS                   off
IdentLookups                    off
User                            nobody
Group                           nobody
AccessDenyMsg                   "Access for %u has been denied"
AuthPAMAuthoritative            on
DeferWelcome                    on
MultilineRFC2228                on
AllowFilter                     "^[a-zA-Z0-9 ,.]*$"
DefaultRoot                     ~ users


# Global Server Context.
<Global>
  DeleteAbortedStores           on
  MaxClients                    3
  MaxLoginAttempts              3
  RateReadBPS                   56400
  RateReadFreeBytes             1440000
  ServerIdent                   on "OpenNA FTP Server ready."
  Umask                         022
</Global>


# Limit normal user logins, because we only want to allow Guest logins.
<Limit LOGIN>
  DenyAll
</Limit>


# Anonymous Server Context.
#
# Anonymous Server Access for GMourani.com
# ----------------------------------------
<Anonymous /home/httpd/gmourani>
  User                          gmourani
  Group                         users
  AnonRequirePassword           on
  Quotas                        on
  QuotaBlockSize                1024
  QuotaBlockName                byte
  QuotaCalc                     on
  QuotaType                     hard
```

```
  DefaultQuota                    25600000
  PathDenyFilter                  "\.quota$"

  <Limit LOGIN>
    AllowAll
  </Limit>

  HideUser                        root
  HideGroup                       root

  <Directory /*>
    AllowOverwrite on
  </Directory>
</Anonymous>
```

This tells the `proftpd.conf` file to set itself up for this particular configuration with:

The General Server Context
The First context in our `ProFTPD` configuration file is the "`General Server Context`". This section is used to encompass everything outside of the other contexts.

`ServerName "OpenNA Linux"`
The "`ServerName`" directive is used to define the string that will be displayed to a user connecting to the server. In our example, users connecting to the `FTP` server will see the "OpenNA Linux" string displayed. You can change the example string for whatever name you would like to be displayed.

`ServerType standalone`
The "`ServerType`" directive is used to configure the server daemon's operating mode. `ProFTPD` can be configured to run as a standalone server or from the `Xinetd` "super server". It is highly recommended to run `ProFTPD` daemon as standalone server and NEVER with `Xinetd`. This is a performance feature.

`DefaultServer on`
The "`DefaultServer`" directive is used to control which server configuration will be used as the default when an incoming `FTP` connection is destined for an `IP` address which is neither the host's primary `IP` address or one of the addresses specified in a `<VirtualHost>` configuration block (if available). When `DefaultServer` is turned `on`, the default server services all unknown destination connections. It's a good idea to enable this option.

`Port 21`
The "`Port`" directive is used to configure the `TCP` port on which the `proftpd` daemon will listen while running in standalone mode. Default `FTP` port number is 21 and we use this value here.

`tcpBackLog 10`
The "`tcpBackLog`" directive is used to control the `TCP` "backlog queue" size connections when listening for `ProFTPD` connections in standalone mode. In other words, when a `TCP` connection is established by the `TCP/IP` stack inside the kernel, there is a short period of time between the actual establishment of the connection and the acceptance of the connection by a user-space program.

The duration of this latency period is widely variable, and can depend upon several factors (hardware, system load, etc). During this period `TCP` connections cannot be accepted, as the port that was previously "listening" has become filled with the new connection. Under heavy connection loads this can result in occasional (or even frequent!) "Connection refused" messages being returned to the incoming client, even when there is a service available to handle requests.

To eliminate this problem, most modern TCP/IP stacks implement a "backlog queue" which is simply a pre-allocation of the resources necessary to handle backlog-size connections during the latency period. The larger the backlog queue, the more connections can be established in a very short time period. This is a performance feature.

```
MaxInstances                    30
```

The "MaxInstances" directive is used to control the maximum number of simultaneous connections allowed on the FTP server. This option can be used to prevent undesirable denial-of-service attacks on the server. If you need to allow more than 30 concurrent connections at once, simply increase this value. This is a security feature.

```
CommandBufferSize               50
```

The "CommandBufferSize" directive is used to control the maximum command length permitted to be sent to the FTP server. This allows us to effectively control what the longest command the FTP server may accept it, and can help protect the FTP server from various Denial of Service or resource-consumption attacks. This is a security feature.

```
UseReverseDNS                   off
```

The "UseReverseDNS" directive is used to prevent the proftpd daemon from attempting to make and check reverse-lookup data connection IP addresses. With ProFTPD, a reverse DNS lookup is performed on the remote host's IP address each time a new connection is made on the FTP server. This checkout may decrease performance of the FTP server and could cause problems when running in chroot mode in some situations. It is highly recommended disabling this feature under ProFTPD by using the 'off' value. This is a performance feature.

```
IdentLookups                    off
```

The "IdentLookups" directive is used to enable or disable the ident protocol (RFC1413) used to attempt to identify the remote username of the FTP connection. In general, ident protocol under FTP is not required and for better performance of your FTP server, it is recommended that this parameter is disabled. This is a performance feature.

```
User                            nobody
```

The "User" directive is used to specify which user the server daemon 'proftpd' will normally run as. In our example, and in most cases, we should use the user called "nobody". This directive instructs the daemon to switch to the specified user as quickly as possible after startup. Never use the super-user 'root' as the user for security reasons. This is a security feature.

```
Group                           nobody
```

The "Group" directive is used to specify which group the server daemon 'proftpd' will normally runs as. In our example, and in most cases, we should use the group called "nobody" again. This directive instructs the daemon to switch to the specified group as quickly as possible after startup. Never use the super-user 'root' as the user group for security reason. This is a security feature.

```
AccessDenyMsg                   "Access for %u has been denied"
```

The "AccessDenyMsg" directive is used to specify which response message must be sent to the FTP client after a failed authentication attempt. Usually, a standard message indicating the reason of failure is sent and in the case of a wrong password, the reason will be "Login incorrect".

To complicate the task for a cracker who tries to access the FTP server, we can return a customized message instead of the standard one like "Access for %u has been denied". In this way, the person doesn't know if the access has been denied for an incorrect login password or something else. This is a security feature.

```
AuthPAMAuthoritative                on
```
The “AuthPAMAuthoritative” directive is used to control whether or not PAM is the ultimate authority on authentication. If this option is set to "on", then other available authentication modules will fail, should PAM authentication fail. Since PAM is the best source for password authentication when SSL is not available, I recommend you use it. This is a security feature.

```
DeferWelcome                        on
```
The “DeferWelcome” directive is used to inform ProFTPD server to not give away any type of information about the host that its daemon is actively running on until a client has successfully authenticated. This is a security feature.

```
MultilineRFC2228                    on
```
The “MultilineRFC2228” directive is used to make “.message” files available into the FTP directory to work with all browsers. You should enable this option if you have any kind of “.message” files available in your FTP directory that you want all browsers to be able to see and/or read.

```
AllowFilter                         "^[a-zA-Z0-9 ,.]*$"
```
The “AllowFilter” directive is used to control what characters may be sent in a command to the FTP server and help to prevent some possible types of attacks against ProFTPD. In our example, we allow only commands containing alphanumeric characters and white space. It is important to note that command filtering is not applied to passwords. This is a security feature.

```
DefaultRoot                         ~ users
```
The “DefaultRoot” directive is used to control the default root directory assigned to a user upon login. One interesting use of this option will be to chroot client into their home directory after authentication. If the magic character "~" is used, then all authenticated clients are automatically placed into their home directory in a chroot environment.

Another interesting argument is the optional group-expression that can be added to the directive to restrict the "DefaultRoot" directive to a UNIX group, groups or subset of groups. In our example, we chroot all authenticated users who are members of the "users" group into their home directories. In this way, we don't have to specify each user in the "DefaultRoot" directive. This is a security feature.

The Global Server Context

The second context in our ProFTPD configuration file is the “Global Server Context”. This section applies universally to all ProFTPD configurations and we use it to set global configurations that will apply to all ProFTPD configurations. A global context configuration begins with <Global> and finishes with </Global>.

```
<Global>
```
The “<Global>” directive opens the block used to define all global configurations that will universally apply to all ProFTPD configurations.

```
DeleteAbortedStores                 on
```
The “DeleteAbortedStores” directive is used to control whether ProFTPD deletes partially uploaded files if the transfer is stopped via the ABOR command. It’s a good idea to enable this option to avoid corrupted files on the FTP server.

```
MaxClients                      3
```
The "MaxClients" directive is used to specify the maximum number of authenticated clients allowed to log into a server or anonymous account. Once this limit is reached, additional clients trying to log into the FTP server will be automatically disconnected.

For a local users FTP server, you can set this value low, and for an anonymous server, you can set this value to the maximum anonymous users allowed to connect to your FTP server depending of your bandwidth. This is a security and optimization feature.

```
MaxLoginAttempts                3
```
The "MaxLoginAttempts" directive is used to configure the maximum number of times a client may attempt to authenticate to the server during a given connection. This is a security feature.

```
RateReadBPS                     56400
```
The "RateReadBPS" directive is used to set the allowed byte per second download bandwidth. Zero means no bandwidth limit. In our example, we set this value to 56400 kbps, meaning that user downloading from the FTP server will download file at 56400 kbps even if they have fast Internet connection. You can use this directive to limit allowed download speed of the FTP server.

```
RateReadFreeBytes               1440000
```
The "RateReadFreeBytes" directive is used to set the amount of bytes to be transferred without any bandwidth limits, so with that option you can give full bandwidth for small files while limiting big ones. In our example, we set this value to 1440000 (1.44 MB), meaning that if user download files under 1.44 MB in size, they will get the full speed of the network and will be limited to the bandwidth limit of 56.4 kbps only if they try to download files bigger than 1.44 MB in size. You can use this directive to control which files size should be downloaded at full speed.

```
ServerIdent                     on "OpenNA FTP Server ready."
```
The "ServerIdent" directive is used to set the default message displayed when a new client connects. Sites desiring to give out minimal information will probably want to enable this option. You can change the example string for whatever you want. This is a security feature.

```
Umask                           022
```
The "Umask" directive is used to define the default mask that should be applied to newly created file and directory permissions within the FTP server. This is a security feature.

```
</Global>
```
The "</Global>" directive closes the block used to define all global configuration that will universally apply to all ProFTPD configurations.

The Limit Server Context

The third context in the ProFTPD configuration file is the "Limit Server Context". This section is used to place limits on whom and how individual FTP commands, or groups of FTP commands, may be used. Here, we use it to limit normal user logins, because we only want to allow Guest logins on this FTP configuration. It's important to note that this directive is defined before any specific user configuration access on the FTP server. This is due to the fact that we have to block any access to the FTP server before allowing specified user's access.

```
<Limit LOGIN>
  DenyAll
</Limit>
```

The above directive, deny all LOGIN access to the FTP server for anyone. For best security, it's recommended to deny everything and only allow what we want further down in the configuration. This is a security feature.

The Anonymous Server Context

The next and last context in the ProFTPD configuration file is the "Anonymous Server Context". It is in this block that we define and allow specific user's access to the FTP server. Some readers may be confused here with the name of this context (Anonymous). They might think that we are trying to define an anonymous FTP access in the configuration, but this is not the case. The anonymous directive could also be used to define local users FTP access.

As you can imagine, we use the same procedure with some modifications to make it work for anonymous FTP access on the server too. See later in this chapter for information on how to configure ProFTPD for anonymous FTP access if you need it. The procedures to define local users FTP access or anonymous FTP server access can be the same with ProFTPD.

```
<Anonymous /home/httpd/gmourani>
```

The "<Anonymous>" directive opens the block used to define all anonymous configurations that will apply to the specified user. An important part here is the addition to the "<Anonymous>" directive of the path where the user FTP home directory resides on the server. In our example, we inform ProFTPD that FTP home directory is located under "/home/httpd/gmourani".

This is useful if you want to provide web site FTP access, since the user is automatically chrooted to their FTP web directory where all their web pages reside. You should change the example FTP home directory "/home/httpd/gmourani" above for whatever you want to use as FTP home directory on the server.

```
User                                    gmourani
```

Contrary to the "User" directive used in the "General Server Context" of ProFTPD, the "User" directive when used in an "Anonymous" block, establishes an anonymous login when a user attempts to login with the specified user ID, as well as permanently switching to the corresponding UID.

This means that the above user, "gmourani", will be the username to use to establish an FTP connection with the FTP server. ProFTPD, will verify if this username "gmourani" is really allowed to connect to the FTP and the home directory "/home/httpd/gmourani" and if the user "gmourani" is really allowed to connect to the FTP, then ProFTPD will runs with the corresponding UID of this user. You should change the example username "gmourani" above for whatever you want to use as a username to connect to the FTP server.

```
Group                                   users
```

Again, and contrary to, the "Group" directive used in the "General Server Context" of ProFTPD, the "Group" directive, when used into an "Anonymous" block, establishes an anonymous login when a user attempts to login with the specified group ID, as well as permanently switching to the corresponding GID.

This means that the above group "users" will be the group name to use to establish the FTP connection with the server. ProFTPD, will verify if the group name "users" is really allowed to connect to its home directory "/home/httpd/gmourani" and if the group "users" is really allowed to connect, then ProFTPD will runs to the corresponding GID of this group. In most cases, you should NOT change the above group name "users". It is a real and existing group name on the system that we use for all GID with ProFTPD. Just keep the default value here.

```
AnonRequirePassword                on
```

OK, here you'll understand why we use the "Anonymous" block of ProFTPD to provide local users FTP access on the server. Normally, anonymous FTP logins do not require the clients to authenticate themselves. Instead, anonymous logins are expected to enter their e-mail addresses when prompted for a password.

Enabling the "AnonRequirePassword" directive requires anonymous logins to enter a valid password, which must match the password of the user that the anonymous daemon runs as. This can be used to create a local users account, which functions exactly as a normal anonymous login does (and thus presents a "chrooted" protected file system to the client), but requires a valid password on the server's host system. Yes, this is the more secure way to allow local users FTP access to the server and the one that we should always use for local user FTP access.

```
Quotas                             on
```

The "Quotas" directive is used to enable or disable FTP quota support with ProFTPD. If the mod_quota module has been compiled with ProFTPD, you will be able to use this useful feature. The "Quotas" directive allows us to use and implement quota limits per user directory without the need to install and use any third party software like quota. If you enable this directive, you will be able to set quota limit per user directory via the ProFTPD configuration file. In the above example we enable quota with ProFTPD.

```
QuotaBlockSize                     1024
```

The "QuotaBlockSize" directive is used to define the block size on which calculations will be made. In our example with set it to 1KB, which equals 1024 bytes.

```
QuotaBlockName                     byte
```

The "QuotaBlockName" directive is used to specify the name to use when reporting quota sizes on the FTP server. In our example, we set it to "byte".

```
QuotaCalc                          on
```

The "QuotaCalc" directive is used to control whether the quota calculation is done on the fly or not. If the directive is set to "on" then the calculation is done on the fly rather than at the end of the FTP session. It's a good idea to enable this option with quota.

```
QuotaType                          hard
```

The "QuotaType" directive is used to define what happens to files which break the quota limits as they are uploaded. Setting the type to "hard" ensures that the file which violates the quota is deleted.

```
DefaultQuota                       25600000
```

The "DefaultQuota" directive is used to set the default quota (in bytes) to be allowed on the web directory or the number of bytes to use as the quota if the user doesn't have a quota file. In our example, we have defined quota and set the limit to 25MB (1024 * 1000 * 25 = 25600000).

```
PathDenyFilter                     "\.quota$"
```

The "PathDenyFilter" directive is used to protect the ".quota" file generated by the FTP server when quota support is enabled. This is a security feature.

```
<Limit LOGIN>
  AllowAll
</Limit>
```

The above directive (context if you prefer), allows all LOGIN access to the FTP server for the specified username of this “Anonymous” block “gmourani”. You’ll remember that we have denied all LOGIN access to the FTP server earlier in this configuration and this is why we now allow FTP access to the specified user here. This is a security feature.

```
HideUser                        root
```

The “HideUser” directive is used to configure a <Directory> or <Anonymous> block (anonymous in our case) to hide all directory entries owned by the specified user, unless the owning user is the currently logged-in, authenticated user. In our example, we hide all possible directories and files entries owned by the super-user ‘root’. This is a security feature.

```
HideGroup                       root
```

The “HideGroup” directive is used to configure a <Directory> or <Anonymous> block (anonymous in our case) to hide all directory entries owned by the specified group, unless the group is the primary group of the currently logged-in, authenticated user. In our example, we hide all possible directories and files entries owned by the super-user ‘root’. This is a security feature.

```
<Directory /*>
  AllowOverwrite on
</Directory>
```

The “AllowOverwrite” directive inside the <Directory> context, permits newly transferred files to overwrite existing ones. By default with ProFTPD, FTP clients cannot overwrite existing files but normally, and inside a user web directory, we want files to be over writeable, therefore turning this option to "on" will let us overwrite existing files on the FTP server.

Used inside a <Directory> context, it can be useful to control which directories on the FTP server can or cannot be overwritten. In our configuration, we allow everything inside the home directory of the user to be overwritten if required.

```
</Anonymous>
```

The “</Anonymous>” directive closes the block used to define the anonymous configuration that applies to the specified user “gmourani”.

/etc/sysconfig/proftpd: The ProFTPD System Configuration File

The /etc/sysconfig/proftpd file is used to specify ProFTPD system configuration information, such as if ProFTPD should run in debug mode. By default the option is disabled and should be enable only for debugging purpose.

- Create the **proftpd** file (touch /etc/sysconfig/proftpd) and add the lines:

```
# Uncomment the following line if you want to debug ProFTPD. All
# log or debug messages will be send to the syslog mechanism.
#
#OPTIONS="-d 5"
```

/etc/pam.d/ftp: The ProFTPD PAM Support Configuration File

For better security of ProFTPD, we have compiled it to use the PAM mechanism for password authentication.

Step 1
To be able to use this feature, we must create the /etc/pam.d/ftp file and add the following parameters inside it.

- Create the **ftp** file (touch /etc/pam.d/ftp) and add the following lines:

```
#%PAM-1.0
auth       required     /lib/security/pam_listfile.so item=user
sense=deny file=/etc/ftpusers onerr=succeed
auth       required     /lib/security/pam_pwdb.so shadow nullok
auth       required     /lib/security/pam_shells.so
account    required     /lib/security/pam_pwdb.so
session    required     /lib/security/pam_pwdb.so
```

Step2
Now, set the permissions of the ftp file to be (0640/-rw-r-----) and owned by the super-user 'root' for security reasons.

- To change the permissions and ownership of the ftp file, use the commands:

```
[root@deep /]# chmod 640 /etc/pam.d/ftp
[root@deep /]# chown 0.0 /etc/pam.d/ftp
```

/etc/ftpusers: The ProFTPD Access Configuration File

This file is used to define a list of users for whom access to the FTP server is always denied. This is a security file where we list all system users that should never get access to the FTP server due to the nature of their UID/GID privileges on the operating system.

Step 1
Please fill free to add to the list below, all users from which you don't want to allow FTP access.

- Create the **ftpusers** file (touch /etc/ftpusers) and add the following lines:

```
root
bin
daemon
sync
mail
nobody
named
rpm
www
amavis
mysql
```

Step 2
Now, set the permissions of the `ftpusers` file to be (`0600/-rw-------`) and owned by the super-user '`root`' for security reason.

- To change the permissions and ownership of the `ftpusers` file, use:
```
[root@deep /]# chmod 600 /etc/ftpusers
[root@deep /]# chown 0.0 /etc/ftpusers
```

`/etc/init.d/proftpd`: The `ProFTPD` Initialization File

The `/etc/init.d/proftpd` script file is responsible to automatically starting and stopping the `ProFTPD` server on your Linux system. Loading the `proftpd` daemon as a standalone daemon will eliminate load time and will even reduce swapping since non-library code will be shared.

Please note that the following script is suitable for Linux operating systems that use `SystemV`. If you Linux system use some other methods like `BSD`, you'll have to adjust the script bellow to make it work for you.

Step 1
Create the **`proftpd`** script file (`touch /etc/init.d/proftpd`) and add the following lines:

```
#!/bin/bash

# This shell script takes care of starting and stopping ProFTPD (FTP server).
#
# chkconfig: 345 85 15
# description: ProFTPD is an enhanced FTP server with a focus toward \
#              simplicity, security, and ease of configuration.
#
# processname: /usr/sbin/proftpd
# config: /etc/sysconfig/network
# config: /etc/proftpd.conf

# Source function library.
. /etc/rc.d/init.d/functions

# Get config.
test -f /etc/sysconfig/network && . /etc/sysconfig/network

if [ -f /etc/sysconfig/proftpd ]; then
      . /etc/sysconfig/proftpd
fi

# Check that networking is up.
[ ${NETWORKING} = "yes" ] || exit 0
[ -f /usr/sbin/proftpd ] || exit 1
[ -f /etc/proftpd.conf ] || exit 1

RETVAL=0

start() {
        echo -n "Starting ProFTPD: "
        daemon proftpd $OPTIONS
        RETVAL=$?
        echo
        touch /var/lock/subsys/proftpd
        return $RETVAL
}

stop() {
```

```
        echo -n "Shutting down ProFTPD: "
        killproc proftpd
        RETVAL=$?
        echo
        rm -f /var/lock/subsys/proftpd
        return $RETVAL
}

restart() {
        stop
        start
}

condrestart() {
        [ -e /var/lock/subsys/proftpd ] && restart
        return 0
}

# See how we were called.
case "$1" in
    start)
        start
        ;;
    stop)
        stop
        ;;
    status)
        status /usr/sbin/proftpd
        ;;
    restart)
        restart
        ;;
    condrestart)
        condrestart
        ;;
    *)
        echo "Usage: proftpd {start|stop|status|restart|condrestart}"
        RETVAL=1
esac
exit $RETVAL
```

Step 2

Once the `proftpd` script file has been created, it is important to make it executable, change its default permissions, create the necessary links and start it. Making this file executable will allow the system to run it, changing its default permissions is to allow only the root user to change this file for security reasons, and the creation of the symbolic links will let the process control initialization of Linux start the program automatically for you at each system reboot.

- To make this script executable and to change its default permissions, use the commands:
  ```
  [root@deep /]# chmod 700 /etc/init.d/proftpd
  [root@deep /]# chown 0.0 /etc/init.d/proftpd
  ```

- To create the symbolic `rc.d` links for `ProFTPD`, use the following commands:
  ```
  [root@deep /]# chkconfig --add proftpd
  [root@deep /]# chkconfig --level 345 proftpd on
  ```

- To start `ProFTPD` software manually, use the following command:
  ```
  [root@deep /]# /etc/init.d/proftpd start
  Starting ProFTPD:        [OK]
  ```

Creating an account for FTP client to connect to the FTP server

Once ProFTPD is running on your server, it's time to create an FTP user account on the system to allow FTP clients to connect the FTP server. Here are the steps to follow each time you want to add a new FTP user to your FTP server.

Step 1
It's important to give to your strictly FTP user no real shell account on the system. This is done because, if for any reason someone successfully gets out of the FTP chrooted environment; they would not have the possibility of using a shell to gain access via other protocols like telnel, ssh, etc.

First, we create new user for this purpose; this user will be the user allowed to connect to your FTP server. This has to be separate from a regular user account with unlimited access because of how the "chroot" environment works. Chroot makes it appear from the user's perspective as if the level of the file system you've placed it in is the top level of the file system.

Here we create a new user called "gmourani" because he's the user that we have used in the proftpd.conf file as an example.

- Use the following command to create a new FTP user. This step must be done for each additional new user you allow to access your FTP server on OpenNA Linux.

```
[root@deep /]# useradd -m -s /bin/false gmourani

[root@deep /]# passwd gmourani
Changing password for user gmourani
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully
```

- Use the following command to create a new FTP user. This step must be done for each additional new user you allow to access your FTP server on Red Hat Linux.

```
[root@deep /]# useradd -g users -s /bin/false gmourani

[root@deep /]# passwd gmourani
Changing password for user gmourani
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully
```

The useradd command will add the new guest user called "gmourani" to our server. The passwd command will set the password for this user "gmourani".

Step2
Once the new user has been added to the system, we have to create their FTP home directory manually. According to the information added into the proftpd.conf file for the user "gmourani", the FTP home directory should be located under /home/httpd/gmourani, therefore we have to create this FTP directory and set the correct permissions on it.

- To create the FTP home directory for user "gmourani", use the following command:
```
[root@deep /]# mkdir /home/httpd/gmourani
```

- To set the correct permissions for the FTP home directory of user "gmourani", use the following command:
```
[root@deep /]# chown -R gmourani.users /home/httpd/gmourani/
```

The mkdir command will create the new FTP home directory for FTP user "gmourani". The chown command will set the correct permissions for this user "gmourani". As you can see, we assume that the GID of user "gmourani" is "users", if this is not the case in your setup, you'll have to change the GID "users" for the one your user has.

NOTE: Don't forget to restart your FTP server for the changes to take effect.

```
[root@deep /]# /etc/init.d/proftpd restart
Shutting down ProFTPD:                [OK]
Starting ProFTPD:                     [OK]
```

Setup an anonymous FTP server

For anonymous FTP server access with ProFTPD, we don't need to create any special directories or even copy binary files or libraries into the anonymous FTP directory to make it work. Anonymous FTP access is really easy to setup with ProFTPD, all you need is to change the default proftpd.conf file and create the required anonymous directory to make it work.

In our example we'll first give anonymous users only access to get files from the FTP anonymous directory on the FTP server and later, show you how to setup ProFTPD to allow anonymous users to upload into a specific subdirectory of the FTP anonymous directory.

Step 1
First, we must create the anonymous directory on our server and change its permissions to allow anonymous FTP access on the server. We decide to create the anonymous directory under the /home directory of the server and name it "ftp".

- To create the anonymous directory named "ftp" with the correct permissions on the server, use the following commands:
```
[root@deep /]# mkdir /home/ftp
[root@deep /]# chown -R ftp.ftp /home/ftp/
```

The above command will create the /home/ftp directory and will change the owner and group of the /home/ftp directory to become the user and group called "ftp". The user and group "ftp" should already exist on your server, remember that we created it previously.

Step2
Once the anonymous directory (`/home/ftp`) has been created with the correct permissions on the server, it time to edit our default `proftpd.conf` file and make the appropriate changes to make it run as an anonymous `FTP` server. Most of the directives are the same as for local users `FTP` access shown previously. We explain only the new directives that do not appear under the local users `FTP` access configuration.

- Edit the **`proftpd.conf`** file (`vi /etc/proftpd.conf`) and set your needs. Below is what we recommend for anonymous `FTP` access:

```
# General Server Context.
ServerName                      "OpenNA Linux"
ServerType                      standalone
DefaultServer                   on
Port                            21
tcpBackLog                      10
MaxInstances                    30
CommandBufferSize               50
UseReverseDNS                   off
IdentLookups                    off
User                            nobody
Group                           nobody
AuthPAMAuthoritative            on
MultilineRFC2228                on
AllowFilter                     "^[a-zA-Z0-9 ,.]*$"

# Global Server Context.
<Global>
  DeleteAbortedStores           on
  MaxClients                    10000
  MaxLoginAttempts              3
  RateReadBPS                   56400
  RateReadFreeBytes             1440000
  ServerIdent                   on "OpenNA FTP Server ready."
  Umask                         022
</Global>

# We don't want normal users logging in at all.
<Limit LOGIN>
  DenyAll
</Limit>

# Normally, we want files to be overwriteable.
<Directory /*>
  AllowOverwrite                on
</Directory>

# A basic Anonymous configuration, no upload directories.
<Anonymous ~ftp>

# Allow Anonymous logins here since all logging are disabled above.
<Limit LOGIN>
  AllowAll
</Limit>

AnonRequirePassword             off
RequireValidShell               off
User                            ftp
Group                           ftp

# We want 'welcome.msg' displayed at login, and '.message' displayed
```

```
# in each newly chdired directory.
DisplayLogin                    welcome.msg
DisplayFirstChdir               .message

# We want clients to be able to login with "anonymous" as well as "ftp".
UserAlias                       anonymous ftp

# Limit the maximum number of anonymous logins.
MaxClients                      10000

# Limit WRITE everywhere in the anonymous chroot jail.
<Limit WRITE>
  DenyAll
</Limit>
</Anonymous>
```

This tells `proftpd.conf` file to set itself up for this particular configuration with:

`<Anonymous ~ftp>`
As we know now, the "`<Anonymous>`" directive (or context if you prefer) opens the block used to define anonymous configurations that will apply to the specified user. The difference between this configuration and the one used earlier for local users access, is that the "`~ftp`" definition that simply informs the `FTP` server to automatically chroot to the users home directory "`/home/ftp`" after successfully connecting.

`AnonRequirePassword off`
Here again, we already know about this directive, but the difference with the previous definition is that we've turned `off` the "`AnonRequirePassword`" directive here, meaning that `ProFTPD` will not ask for a valid password and will accept e-mail address or whatever users want to enter when prompted for a password.

`RequireValidShell off`
The "`RequireValidShell`" directive is used to configure the server, anonymous login to allow or deny logins which do not have a shell binary listed in `/etc/shells`. By default, `ProFTPD` disallows logins if the user's default shell is not listed in `/etc/shells`. For anonymous access, we must turn the default setting of "`on`" to become "`off`" or the anonymous `FTP` connection will not work.

`User ftp`
Here, the configuration is the same as for the previous configuration for local users with a difference that the user specified here is the anonymous user under which the `FTP` daemon will run. You should NOT change the above user since it is the default `ProFTPD` user used by anonymous `FTP` connections.

`Group ftp`
The same is true for the "`Group`" directive. We should define and keep "`ftp`" as the anonymous user for `ProFTPD` to work in an anonymous configuration.

`DisplayLogin welcome.msg`
The "`DisplayLogin`" directive is used to specify an `ASCII` text filename (`welcome.msg`), which will be displayed to the user when they initially login to the anonymous `FTP` server. The file is searched for in the initial directory a user is placed in immediately after login (anonymous-root `/home/ftp` directory for anonymous logins).

```
DisplayFirstChdir                .message
```
The "DisplayFirstChdir" directive is used to configure an ASCII text filename (.message), which will be displayed to the user the first time they change, into a directory (via CWD) per a given session. The file will also be displayed if ProFTPD detects that its last modification time has changed since the previous CWD into a given directory. The file is searched for inside all accessible directories of the anonymous-root /home/ftp directory. You can use as many ASCII text filename (.message) as you want in each directory.

```
UserAlias                        anonymous ftp
```
ProFTPD requires a real username/uid when authenticating users. There are however times when additional aliases are required, but it is undesirable to provide additional login accounts to achieve it. With an anonymous FTP server configuration, it is normal for the server to use "ftp" as the primary authentication user, however it is common practice for FTP users to login into the anonymous FTP server using "anonymous" as username. This is achieved by adding the above lines to the configuration file.

```
MaxClients                       10000
```
The "MaxClients" directive is used to specify the maximum number of authenticated clients allowed logging into a server or anonymous account. Once this limit is reached, additional clients trying to log into the FTP server will be automatically disconnected.

For a local users FTP server, you can set this value low, and for an anonymous server, you can set this value to the maximum anonymous users allowed to connect to your FTP server depending of your bandwidth. Here we set the value to allow a maximum of 10000 FTP clients to connect to the anonymous FTP server. This is a security and optimization feature.

```
<Limit WRITE>
  DenyAll
</Limit>
```
The above directive (or context if you prefer), deny all WRITE access to the FTP server for everyone. This is required for security reasons when configuring ProFTPD to run as an anonymous FTP server. We don't want to let anonymous users write and change files in the anonymous FTP area. This is a security feature.

```
</Anonymous>
```
The "</Anonymous>" directive closes the block used to define all anonymous configurations that apply to the specified user "ftp".

NOTE: Don't forget to restart your FTP server for the changes to take effect.

```
[root@deep /]# /etc/init.d/proftpd restart
Shutting down ProFTPD:                [OK]
Starting ProFTPD:                     [OK]
```

Allow anonymous users to upload to the FTP server

Once our configuration for an anonymous FTP access is running properly, we can decide to allow anonymous users to upload on the anonymous FTP server subdirectory of our choice. Below are the steps to follow if you want to allow anonymous users to be able to upload to your anonymous FTP server.

Step 1
We have to create a subdirectory inside the existing anonymous directory on our server and change its permissions to allow anonymous upload into this subdirectory of the FTP server. For this example, we decide to name the upload subdirectory "uploads".

- To create the upload subdirectory called "uploads" with the correct permission mode on the server, use the following commands:

```
[root@deep /]# mkdir /home/ftp/uploads
[root@deep /]# chown -R ftp.ftp /home/ftp/uploads/
```

The above command will create the /home/ftp/uploads subdirectory and will change the owner and group of the /home/ftp/uploads subdirectory to become the user and group called "ftp".

Step2
Once the upload subdirectory (/home/ftp/uploads) has been created with the correct permissions on the server, it time to edit our default anonymous proftpd.conf file and make the changes to allow uploading files inside the anonymous FTP server.

Here are the required directives to add to your default anonymous proftpd.conf file. Most directives are the same as for the anonymous configuration FTP access. We'll explain only the new directives that do not appear under the anonymous FTP access configuration. Text in bold is what we've added to the default anonymous configuration file.

- Edit the **proftpd.conf** file (vi /etc/proftpd.conf) and set your needs. Below is what we recommend you for anonymous FTP access with upload capability:

```
# General Server Context.
ServerName                      "OpenNA Linux"
ServerType                      standalone
DefaultServer                   on
Port                            21
tcpBackLog                      10
MaxInstances                    30
CommandBufferSize               50
UseReverseDNS                   off
IdentLookups                    off
User                            nobody
Group                           nobody
AuthPAMAuthoritative            on
MultilineRFC2228                on
AllowFilter                     "^[a-zA-Z0-9 ,.]*$"

# Global Server Context.
<Global>
  DeleteAbortedStores           on
  MaxClients                    10000
  MaxLoginAttempts              3
  RateReadBPS                   56400
  RateReadFreeBytes             1440000
```

```
  ServerIdent                     on "OpenNA FTP Server ready."
  Umask                           022
</Global>

# We don't want normal users logging in at all.
<Limit LOGIN>
  DenyAll
</Limit>

# Normally, we want files to be overwriteable.
<Directory /*>
  AllowOverwrite                  on
</Directory>

# A basic Anonymous configuration, no upload directories.
<Anonymous ~ftp>

# Allow Anonymous logins here since all logging are disabled above.
<Limit LOGIN>
  AllowAll
</Limit>

AnonRequirePassword               off
RequireValidShell                 off
User                              ftp
Group                             ftp

# We want 'welcome.msg' displayed at login, and '.message' displayed
# in each newly chdired directory.
DisplayLogin                      welcome.msg
DisplayFirstChdir                 .message

# We want clients to be able to login with "anonymous" as well as "ftp".
UserAlias                         anonymous ftp

# Limit the maximum number of anonymous logins.
MaxClients                        10000

# Limit WRITE everywhere in the anonymous chroot jail.
<Limit WRITE>
  DenyAll
</Limit>

# Upload directory that allows storing files but not retrieving
# or creating directories.
<Directory uploads/*>
  HiddenStor on
  <Limit READ RMD DELE MKD>
    DenyAll
  </Limit>

  <Limit STOR CWD>
    AllowAll
  </Limit>
</Directory>
</Anonymous>
```

This tells the `proftpd.conf` file to set itself up for this particular configuration with:

`<Directory uploads/*>`
The "`<Directory>`" directive opens a new context used to define anonymous upload configurations that will apply to the specified subdirectory "`/home/ftp/uploads`" and any files inside this subdirectory.

`HiddenStor on`
The "`HiddenStor`" directive if turned "`on`" will protect files uploaded to the `FTP` server by providing a degree of atomicity. This is accomplished by preventing incomplete uploads and files being used while they're still in the progress of being uploaded. It's a good idea to enable this option.

```
<Limit READ RMD DELE MKD>
  DenyAll
</Limit>
```
Here we use the "`Limit`" configuration block to place access restrictions on `FTP` commands, within a given context. The `FTP` commands that we restrict here are the "`READ`", "`RMD`", "`DELE`", and "`MKD`" commands meaning that all `FTP` commands which deal with file or directory reading, removing, deleting, or creating will be denied for all users. This is a security feature.

```
<Limit STOR CWD>
  AllowAll
</Limit>
```
Here we use another "`Limit`" configuration block to place access right on two `FTP` commands, within a given context. The `FTP` commands that we allow here are the "`STOR`" and "`CWD`" commands meaning that files transferred from the `FTP` client to the `FTP` server will be allowed for all users. This is the part of the configuration that makes file uploads possible to the "`uploads`" subdirectory of the anonymous `FTP` server area and provides optimum security. This is a security feature.

`</Directory>`
The "`</Directory>`" directive close the context used to define anonymous upload configuration that apply to the specified subdirectory "`/home/ftp/uploads`".

NOTE: Don't forget to restart your `FTP` server for the changes to take effect.

```
[root@deep /]# /etc/init.d/proftpd restart
Shutting down ProFTPD:              [OK]
Starting ProFTPD:                   [OK]
```

Running ProFTPD with SSL support

This section applies only if you want to run ProFTPD through SSL. By default, ProFTPD does not support SSL and we have to compile it with an external patch to enable the TLS extensions with the FTP protocol. The patch that allows us to this is available from the following FTP site: ftp://ftp.runestig.com/pub/proftpd-tls/.

You have to download the version number equal to the ProFTPD version number in order for SSL support to work. At the beginning of this chapter, we patched the software with the TLS extensions patch, therefore, we only need to create the required certificate files and reconfigure our proftpd.conf file to enable SSL support.

Creating the necessary ProFTPD certificate keys:

Below we'll show you how to create a certificate or a self-signed certificate with your own CA certificate for ProFTPD. The principle is exactly the same as for creating a certificate or a self-signed certificate for a Web Server. We'll assume that your own CA certificates have been already created, if this is not the case, please refer to OpenSSL chapter for further information.

Step 1
Here, we have to create a new FTP certificate for ProFTPD. This FTP certificate becomes our private key and doesn't need to be encrypted. This is required for an unattended startup of ProFTPD; otherwise you will have to enter the pass phrase each time ProFTPD is started.

- To create a certificate private key without a pass phrase, use the following command:

```
[root@deep /]# cd /usr/share/ssl
[root@deep ssl]# openssl genrsa -rand
random1:random2:random3:random4:random5 -out ftpd-rsa-key.pem 1024
22383 semi-random bytes loaded
Generating RSA private key, 1024 bit long modulus
.++++++
.................................++++++
e is 65537 (0x10001)
```

NOTE: The name of our certificate private key for ProFTPD is "ftpd-rsa-key.pem", this is important because we cannot use any other name. If you try to create a private certificate with a different name than the one we use above, the FTP server will simply not recognize your certificate private key. Therefore it is very important to use "ftpd-rsa-key.pem" as the name of this certificate private key.

Step 2
Once the private key has been made, we must generate a **C**ertificate **S**igning **R**equest (CSR) with the server RSA private key. The command below will prompt you for the X.509 attributes of your certificate. If you prefer to have your Certificate Signing Request (CSR) signed by a commercial **C**ertifying **A**uthority (CA) like Thawte or Verisign you need to post the CSR file that will be generated below into a web form, pay for the signing, and await the signed Certificate.

- To generate the CSR, use the following command:

```
[root@deep ssl]# openssl req -new -key ftpd-rsa-key.pem -out ftpd-rsa-
csr.pem
Using configuration from /usr/share/ssl/openssl.cnf
You are about to be asked to enter information that will be incorporated
into your certificate request.
```

```
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [CA]:
State or Province Name (full name) [Quebec]:
Locality Name (eg, city) [Montreal]:
Organization Name (eg, company) [OpenNA.com FTP Server]:
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) [ftp.openna.com]:
Email Address [noc@openna.com]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

WARNING: Be sure that you've entered the FQDN (**F**ully **Q**ualified **D**omain **N**ame) of the FTP Server when OpenSSL prompts you for the "**Common Name**".

Step 3
This step is needed only if you want to sign as your own CA the csr certificate key. Now we must sign the new certificate with our own certificate authority that we have already created for generation of the Web Server certificate under the OpenSSL chapter (ca.crt). If the self signed CA certificate doesn't exist, then refer to the chapter related to OpenSSL for more information about how to create it.

- To sign with our own CA, the csr certificate, use the following command:

```
[root@deep ssl]# /usr/share/ssl/misc/sign ftpd-rsa-csr.pem
CA signing: ftpd-rsa-csr.pem -> ftpd-rsa-csr.pem.crt:
Using configuration from ca.config
Enter PEM pass phrase:
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName           :PRINTABLE:'CA'
stateOrProvinceName   :PRINTABLE:'Quebec'
localityName          :PRINTABLE:'Montreal'
organizationName      :PRINTABLE:'OpenNA.com FTP Server'
commonName            :PRINTABLE:'ftp.openna.com'
emailAddress          :IA5STRING:'noc@openna.com'
Certificate is to be certified until Feb 21 11:36:12 2003 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
CA verifying: ftpd-rsa-csr.pem.crt <-> CA cert
ftpd-rsa-csr.pem.crt: OK
```

WARNING: If you receive an error message saying that the csr certificate that you are trying to sign already exists, it is because the information you have entered during the generation of the certificate key is the same as another, which you have already created. In this case you must at least, change one bit of information in the new certificate key you want to create before signing the certificate with your own CA.

Step 4
Once our certificate has been signed, we must rename it for the FTP server to be able to recognize and use it. Remember that ProFTPD requires that certificates have a specific name to be able to use them.

- To properly rename your certificate, use the following command:
```
[root@deep ssl]# mv ftpd-rsa-csr.pem.crt ftpd-rsa.pem
```

Step 5
Finally, we must place the certificates files (ftpd-rsa-key.pem and ftpd-rsa.pem) to the appropriate directories for ProFTPD to be able to find them when it starts on the system.

- To place the certificates into the appropriate directory, use the following commands:
```
[root@deep ssl]# mv ftpd-rsa-key.pem private/
[root@deep ssl]# mv ftpd-rsa.pem certs/
[root@deep ssl]# chmod 400 private/ftpd-rsa-key.pem
[root@deep ssl]# chmod 400 certs/ftpd-rsa.pem
[root@deep ssl]# rm -f ftpd-rsa-csr.pem
```

With the above commands, we move the "ftpd-rsa-key.pem" file to the /private directory and the "ftpd-rsa.pem" file to the /certs directory. After that we change the permissions of both certificates to be only readable by the super-user 'root' for security reasons and remove the "ftpd-rsa-csr.pem" file from our system since it is no longer needed.

Adding the required SSL parameters to the 'proftpd.conf' file:

Once the ProFTPD certificates have been created and moved to the appropriate location, we must add some new directives into the proftpd.conf file for ProFTPD to be configured to run with SSL support.

Step 1
Below we show you the directives to add to your default proftpd.conf file for ProFTPD to run with SSL support. Text in bold is what we have added to the default ProFTPD configuration file. Remember that SSL support with the FTP server is required only when you run your FTP server for local users access, there is really no need or reason to run an FTP server with SSL support on an anonymous FTP configuration since an anonymous FTP server does not ask you to enter a valid password to connect to the FTP server. I assume that this is clear for everyone but I prefer to repeat it.

- Edit your **proftpd.conf** file (vi /etc/proftpd.conf), and add the following directives inside the file to enable SSL support.

```
# General Server Context.
  ServerName                    "OpenNA Linux"
```

```
  ServerType                    standalone
  DefaultServer                 on
  Port                          990
  tcpBackLog                    10
  MaxInstances                  30
  CommandBufferSize             50
  UseReverseDNS                 off
  IdentLookups                  off
  User                          nobody
  Group                         nobody
  AccessDenyMsg                 "Access for %u has been denied"
  AuthPAMAuthoritative          on
  DeferWelcome                  on
  MultilineRFC2228              on
  AllowFilter                   "^[a-zA-Z0-9 ,.]*$"
  DefaultRoot                   ~ users

# TLS related options.
  TlsRsaCertFile                ftpd-rsa.pem
  TlsRsaKeyFile                 ftpd-rsa-key.pem
  TlsCipherList
 ALL:!ADH:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP
  TlsRequired                   on
# Verify any peer certificates.
  TlsCertsOk                    on


# Global Server Context.
<Global>
  DeleteAbortedStores           on
  MaxClients                    3
  MaxLoginAttempts              3
  RateReadBPS                   56400
  RateReadFreeBytes             1440000
  ServerIdent                   on "OpenNA FTP Server ready."
  Umask                         022
</Global>


# Limit normal user logins, because we only want to allow Guest logins.
<Limit LOGIN>
  DenyAll
</Limit>


# Anonymous Server Context.
#
# Anonymous Server Access for GMourani.com
# ----------------------------------------
<Anonymous /home/httpd/gmourani>
  User                          gmourani
  Group                         users
  AnonRequirePassword           on
  Quotas                        on
  QuotaBlockSize                1024
  QuotaBlockName                byte
  QuotaCalc                     on
  QuotaType                     hard
  DefaultQuota                  25600000
  PathDenyFilter                "\.quota$"

  <Limit LOGIN>
    AllowAll
```

```
  </Limit>

  HideUser                      root
  HideGroup                     root

  <Directory /*>
    AllowOverwrite on
  </Directory>
</Anonymous>
```

This tells the `proftpd.conf` file to set itself up for this particular configuration with:

`Port 990`
The "`Port`" directive is used to inform the `FTP` server on which port it should listen. On normal `FTP` operations, we set this directive to `21`, which is the official port of the `FTP` protocol. When configuring `FTP` to run with `SSL` support, we have to change the default port of `21` to become `990` since `FTP` with `SSL` support runs on this port by default and `FTP` clients expect to find `FTP` with `SSL` support on this port too.

`TlsRsaCertFile ftpd-rsa.pem`
The "`TlsRsaCertFile`" directive is used to specify the name of the cert certificate key file on the server. The default location of this certs certificate key is supposed to be under the `/usr/share/ssl/certs` directory. Never change the name of this certificate because `ProFTPD` cannot use another name.

`TlsRsaKeyFile ftpd-rsa-key.pem`
The "`TlsRsaKeyFile`" directive is used to specify the name of the private certificate key file on the server. The default location of this private certificate key is supposed to be under the `/usr/share/ssl/private` directory. Again, never change the name of this certificate because `ProFTPD` cannot use another name.

`TlsCipherList ALL:!ADH:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP`
The "`TlsCipherList`" directive is used to specify the list of supported cipher algorithms on this `FTP` server. The above list enables all available ciphers with exception of the `ADH` key exchange.

`TlsRequired on`
The "`TlsRequired`" directive is used to control if an `SSL` connection is the only way to connect to the `FTP` server or not. If we set this option to "`on`", then an `SSL` connection will be required to connect to the `FTP` server. If we set this option to "`off`", then you will be able to connect to the `FTP` server with or without an `SSL` connection.

`TlsCertsOk on`
The "`TlsCertsOk`" directive is used to inform the `FTP` server if it should verify any peer client certificates or not. If we set this option to "`on`", then any peer certificates will not be verified and the `FTP` server will assume that client peer certificates (if available) are ok.

NOTE: Don't forget to restart your `FTP` server for the changes to take effect.

```
[root@deep /]# /etc/init.d/proftpd restart
Shutting down ProFTPD:              [OK]
Starting ProFTPD:                   [OK]
```

Securing ProFTPD

This section deals specifically with actions we can take to improve and tighten security under ProFTPD. Once properly compiled, installed, and configured, there are only some little things that we can do to better secure it. Most of the important security measures are already made within the software.

Controlling clients access to the FTP server:

In some situations, we need to control access to the FTP server. For example, we may need to restrict FTP connections to our private network and/or some IP addresses only. This is possible with the following directives:

```
<Limit LOGIN>
  Order Allow,Deny
  Allow from 207.35.78.,192.168.1.
  Deny from all
</Limit>
```

The above option "Allow" is used inside a <limit> context to explicitly specify which hosts and/or networks have access to the commands or operations being limited (in our example LOGIN). This directive is used to create precise access control rules such as we can do for a web server like Apache. In our example we specify the order in which we want this directive to be applied "Order Allow,Deny", then we allow two distinct ranges of IP addresses to log in "Allow from 207.35.78.,192.168.1.", and finally deny all other IP addresses to be able to log in "Deny from all".

Step1

If you want to implement this kind of access control into your proftpd.conf configuration file, then add the above lines into the appropriate place inside your ProFTPD configuration. Don't forget to change the example IP address(es) for the one that you have.

- Edit your original **proftpd.conf** file (vi /etc/proftpd.conf) and add the following lines. You have to change the example parameters to reflect your own settings.

```
# General Server Context.
ServerName                      "OpenNA Linux"
ServerType                      standalone
DefaultServer                   on
Port                            21
tcpBackLog                      10
MaxInstances                    30
CommandBufferSize               50
UseReverseDNS                   off
IdentLookups                    off
User                            nobody
Group                           nobody
AccessDenyMsg                   "Access for %u has been denied"
AuthPAMAuthoritative            on
DeferWelcome                    on
MultilineRFC2228                on
AllowFilter                     "^[a-zA-Z0-9 ,.]*$"
DefaultRoot                     ~ users


# Global Server Context.
<Global>
  DeleteAbortedStores           on
  MaxClients                    3
  MaxLoginAttempts              3
```

```
  RateReadBPS                     56400
  RateReadFreeBytes               1440000
  ServerIdent                     on "OpenNA FTP Server ready."
  Umask                           022
</Global>


# Limit normal user logins, because we only want to allow Guest logins.
<Limit LOGIN>
  DenyAll
</Limit>


# Anonymous Server Context.
#
# Anonymous Server Access for GMourani.com
# ----------------------------------------
<Anonymous /home/httpd/gmourani>
  User                            gmourani
  Group                           users
  AnonRequirePassword             on
  Quotas                          on
  QuotaBlockSize                  1024
  QuotaBlockName                  byte
  QuotaCalc                       on
  QuotaType                       hard
  DefaultQuota                    25600000
  PathDenyFilter                  "\.quota$"

  <Limit LOGIN>
    Order Allow,Deny
    Allow from 207.35.78.,192.168.1.
    Deny from all
  </Limit>

  HideUser                        root
  HideGroup                       root

  <Directory /*>
    AllowOverwrite on
  </Directory>
</Anonymous>
```

ProFTPD Administrative Tools

The commands listed below are some of the most used, but many more exist. Check the manual pages for more details.

ftpwho

The `ftpwho` program utility displays all active `ftp` users, and their current process information on the system.

- To displays all active `ftp` users and their current process, use the following command:

```
[root@deep /]# ftpwho
Master proftpd process 11570:
 4798 0m3s   proftpd: gmourani - 127.0.0.1: anonymous: IDLE
Service class                      -   1 user
```

ftpcount

The `ftpcount` program utility, which is a simplified version of `ftpwho`, shows only the current number of users logged into the system.

- To shows only the current number of users logged in to the system, use the command:

```
[root@deep /]# ftpcount
Master proftpd process 11570:
Service class                      -   1 user
```

Further documentation

For more details, there are some manual pages about `ProFTPD` that you could read:

```
$ man proftpd (8)              - Professional configurable, secure  FTP server.
$ man ftpshut (8)              - Shut down all proftpd servers at a given time.
$ man xferlog (5)              - ProFTPD server logfile.
$ man ftpcount (1)             - Show current number of connections.
$ man ftpwho (1)               - show current process information.
```

CHAPTER 42

vsFTPd

IN THIS CHAPTER

1. Compiling - Optimizing & Installing `vsFTPd`
2. Configuring `vsFTPd`
3. Creating an account for `FTP` client to connect to the `FTP` server
4. Setup an anonymous `FTP` server
5. Allow anonymous users to upload to the `FTP` server

Linux vsFTPd

Abstract

vsFTPd stand for "**v**ery **s**ecure **FTP d**aemon", it is an FTP server that has been written from scratch with security and speed as primary goals. It provides most of the necessary features that you could expect from a modern FTP server. If you are looking for security, performance, and stability, then vsFTPd is for you.

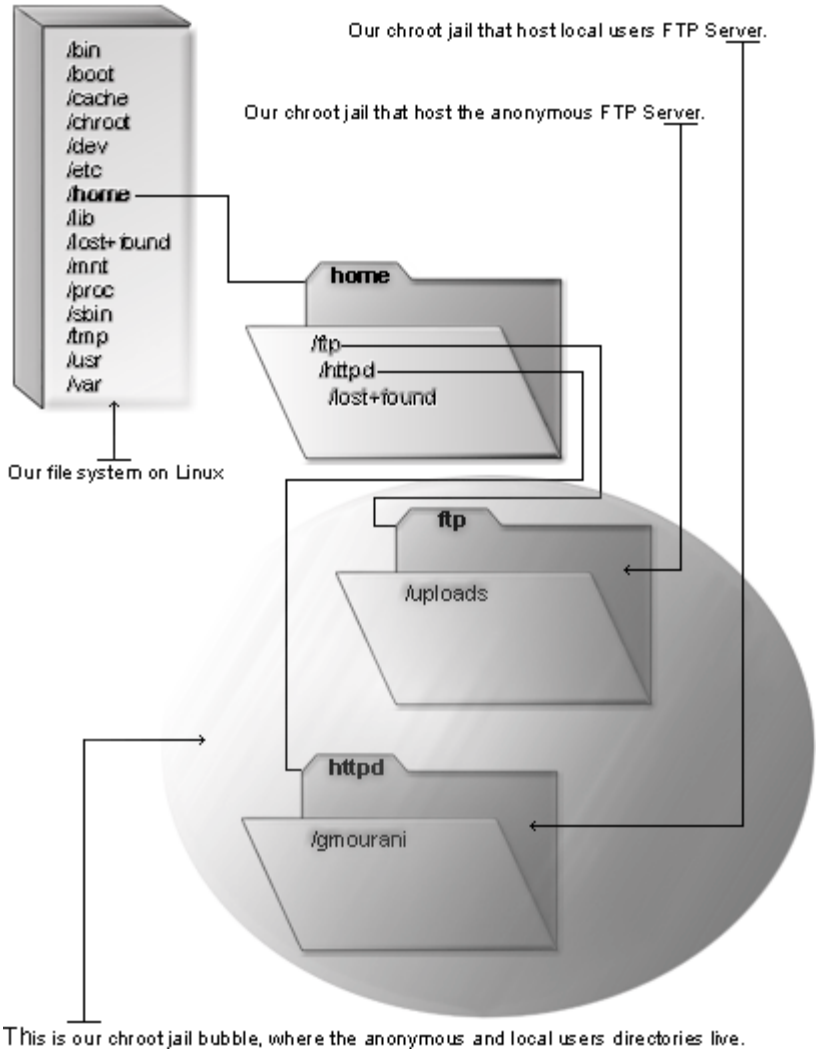
The primary goal of vsFTPd is to provide highly secure FTP server, it is really easy to compile, install and configure. If you are new in the world of FTP servers, then I recommend you to go with vsFTPd. If you are looking for a complete and configurable FTP server providing many useful features for web hosting, then ProFTPD is the one to go with. vsFTPd is the perfect FTP server to offer anonymous FTP access.

The only thing that I don't like with it, is that it does not allow us to run it as a standalone daemon server, we can only run it via another daemon that accepts incoming connections such as inetd, Xinetd or tcpserver. In our installation, we will use tcpserver to make it run on the system. I don't want to use inetd which is too old and buggy, or even Xinetd which is simply not acceptable when we want to provide a very fast, secure and workable FTP server for our system.

The tcpserver software from Dan Bernstein's is the best available method to run vsFTPd on Linux, since is it the fastest and most secure super server on the Internet. It code is well-written and very powerful to use. Yes, one of the best UNIX programming style codes that we can see today.

In this chapter, we will show and explain to you how to install, configure, and run vsFTPd for local user FTP connections and for anonymous FTP connections. As usual, we will begin our configuration with an example suitable for FTP server access for local users and will continue with an example suitable for an anonymous FTP server access.

vsFTPd running in chroot jail



These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest vsFTPd version number is 1.0.1

The procedures given in this chapter are likely to work on all Linux platforms, but we have only tested it on OpenNA Linux and Red Hat Linux.

Packages

The following are based on information as listed by vsFTPd as of 2002/04/19. Please regularly check http://vsftpd.beasts.org/ for the latest status. We chose to install the required component from source file because it provides the facility to fine tune the installation.

Source code is available from:
vsFTPd Homepage: http://vsftpd.beasts.org/
vsFTPd FTP Site: 163.1.18.131
You must be sure to download: vsftpd-1.0.1.tar.gz

Prerequisites

vsFTPd requires that the software below is already installed on your system to be able to compile successfully. If this is not the case, you must install it. Please make sure you have this program installed on your machine before you proceed with this chapter.

- ✓ ucspi-tcp is required to run vsFTPd on your system.

NOTE: For more information on the ucspi-tcp software, see its related chapter in this book.

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all installed files into the system if the program is updated in the future. To solve this problem, it is a good idea to make a list of files on the system before you install vsFTPd, and one afterwards, and then compares them using the **diff** utility to find out what files are placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep root]# find /* > vsFTPd1
  ```
- And the following one after you install the software:
  ```
  [root@deep root]# find /* > vsFTPd2
  ```
- Then use the following command to get a list of what changed:
  ```
  [root@deep root]# diff vsFTPd1 vsFTPd2 > vsFTPd-Installed
  ```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. We use the /root directory of the system to stock all generated list files.

Compiling - Optimizing & Installing vsFTPd

Below are the steps that you must make to configure, compile and optimize the vsFTPd software before installing it into your Linux system. First off, we install the program as user 'root' so as to avoid authorization problems.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- This can be done with the following commands:

```
[root@deep /]# cp vsftpd-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf vsftpd-version.tar.gz
[root@deep tmp]# cd vsftpd-version
```

Step 2
vsFTPd cannot run as super-user root; for this reason we must create a special user with no shell privileges on the system for running vsFTPd daemon.

- To create this special vsFTPd user on OpenNA Linux, use the following command:

```
[root@deep vsftpd-1.0.1]# groupadd -g 24 ftp > /dev/null 2>&1 || :
[root@deep vsftpd-1.0.1]# useradd -c "FTP Server" -d /home/ftp -g 24 -s
/bin/false -u 24 ftp > /dev/null 2>&1 || :
```

- To create this special vsFTPd user on Red Hat Linux, use the following command:

```
[root@deep vsftpd-1.0.1]# groupadd -g 24 ftp > /dev/null 2>&1 || :
[root@deep vsftpd-1.0.1]# useradd -u 24 -g 24 -s /bin/false -M -r -d
/home/ftp ftp > /dev/null 2>&1 || :
```

The above commands will create a null account, with no password, no valid shell, no files owned-nothing but a UID and a GID for the program. Remember that the vsFTPd daemon does not need to have a shell account on the server.

Step 3
Now, edit the **shells** file (vi /etc/shells) and add a non-existent shell name "/bin/false", which is the one we used in the useradd command above.

```
[root@deep vsftpd-1.0.1]# vi /etc/shells
/bin/bash2
/bin/bash
/bin/sh
/bin/false ← This is our added no-existent shell
```

Step 4
It is time to compile, optimize, build and install vsFTPd for our system. Unlike the commands we use to compile other software in this book, with vsFTPd we only need to use the "make" command for the software to be installed on the system.

Therefore, we build vsFTPd with the 'make' command and produce a list of files on the system before we install the software, and one afterwards, then compare them using the **diff** utility to find out what files are placed where and finally install vsFTPd.

- To compile, optimize, build and install vsFTPd use the following commands:

```
[root@deep vsftpd-1.0.1]# cd
[root@deep root]# find /* > vsFTPd1
[root@deep root]# cd /var/tmp/vsftpd-1.0.1/
[root@deep vsftpd-1.0.1]# make CFLAGS="-O2 -march=i686 -funroll-loops"
[root@deep vsftpd-1.0.1]# make install
[root@deep vsftpd-1.0.1]# chmod 0511 /usr/sbin/vsftpd
[root@deep vsftpd-1.0.1]# rm -f /etc/xinetd.d/vsftpd
[root@deep vsftpd-1.0.1]# cd
[root@deep root]# find /* > vsFTPd2
[root@deep root]# diff vsFTPd1 vsFTPd2 > vsFTPd-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

Step 5
Once the compilation, optimization and installation of the software have been finished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete vsFTPd and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf vsftpd-version/
[root@deep tmp]# rm -f vsftpd-version.tar.gz
```

The rm command as used above will remove all the source files we have used to compile and install vsFTPd. It will also remove the vsFTPd compressed archive from the /var/tmp directory.

Configuring vsFTPd

After vsFTPd has been built and installed successfully in your system, your next step is to configure and customize its configuration files to fit your needs.

- ✓ /etc/vsftpd.conf (The vsFTPd Configuration File)
- ✓ /etc/pam.d/ftp (The vsFTPd PAM Support Configuration File)
- ✓ /etc/ftpusers (The vsFTPd Access Configuration File)
- ✓ /etc/logrotate.d/vsftpd (The vsFTPd Log rotation File)
- ✓ /etc/init.d/vsftpd (The vsFTPd Initialization File)

/etc/vsftpd.conf: The vsFTPd Configuration File

The /etc/vsftpd.conf file is the main configuration file for vsFTPd. It is in this configuration file that vsFTPd gets all of its information and the way it should run on your system. We can configure the vsftpd.conf file to run vsFTPd as an anonymous FTP server, or as local users FTP server for web hosting, etc.

Step 1
Different configurations exist, and we will show you later how to configure it to run as a local users FTP server and also as an anonymous FTP server on your system. We start our configuration by showing you how to configure it to run as a local users FTP server.

- Create the **vsftpd.conf** file (touch /etc/vsftpd.conf). Below is what we recommend you for a local users FTP access:

```
anonymous_enable=NO
local_enable=YES
write_enable=YES
local_umask=022
xferlog_enable=YES
connect_from_port_20=NO
one_process_model=NO
nopriv_user=ftp
ftpd_banner=OpenNA Linux FTP server ready.
chroot_local_user=YES
```

This tells vsftpd.conf file to set itself up for this particular configuration with:

anonymous_enable=NO
The "anonymous_enable" option is used to control whether anonymous logins are permitted on the FTP server. If you set this option to "YES", then anonymous logins will be allowed. The default parameter for this option is "YES" and since we are configuring vsFTPd to run as a local users FTP server in this configuration file, we have to disable this option by saying "NO" here.

local_enable=YES
The "local_enable" option is used to control whether local logins are permitted on the FTP server. If you set this option to "YES", then local logins or if you prefer, local users FTP access will be allowed. The default parameter for this option is "NO" and since we are configuring vsFTPd to run for local users FTP access in this configuration file, we have to enable this option by saying "YES" here. If you don't enable this option, you'll not be able to log in to the FTP server.

write_enable=YES
The "write_enable" option is used to control whether any FTP commands related to file system changes would be allowed on the FTP server. For a local users FTP access, we must enable this option by saying "YES" here to allow users to manage their FTP home directory. For an anonymous connection where anonymous users are only allowed to download files from the FTP server, we'll turn this option off by saying "NO" here. Since we are configuring vsFTPd for local users FTP access, we must say "YES" here.

local_umask=022
The "local_umask" option is used to set the default umask value to use for file creation on the FTP server. The value you enter here will represent the permission mode of newly created files for local users on the FTP server. The value of "022" allow new files to be created on the FTP server with mode permission set to 0644 (-rw-r--r--), which is a safe mode. This is a security feature.

xferlog_enable=YES
The "xferlog_enable" option is used to control whether we want to maintain uploading and downloading of files from the FTP server into a log file called vsftpd.log located under the /var/log directory. It's a good idea to enable this option by saying "YES" here. This is a security feature.

connect_from_port_20=NO
The "connect_from_port_20" option is used to control whether PORT style data connections use port 20 (ftp-data) on the FTP server. For security reasons, some clients may insist that this is the case. Conversely, disabling this option enables vsftpd to run with slightly less privileges. In our configuration, we disable this option by saying "NO" here. Try to run your FTP server with this option disables and if you encounter some problem then enable it. This is a security feature.

one_process_model=NO
The "one_process_model" option allows us to use a different security model with the FTP server. This option instructs vsFTPd to only use one process per connection. By default, vsFTPd uses two processes per connection to run, and on highly loaded FTP sites, this can penalize performance. Therefore, if you FTP server support huge numbers of simultaneously connected users, you may need to enable this option otherwise you should keep the default setting of "NO" here. Only enable this option for highly loaded FTP server. This is a performance feature.

nopriv_user=ftp
The "nopriv_user" option is used to specify the name of the user that is used by the vsftpd daemon when it wants to be totally unprivileged. Here we define the user called "ftp" that we have created previously in this chapter. Remember this "ftp" user have a null account, with no password, no valid shell, no files owned-nothing but a UID and a GID. This is a security feature.

ftpd_banner=OpenNA Linux FTP server ready.
The "ftpd_banner" option is used to set the default message displayed when a new client connects to the FTP server. Sites desiring to give out minimal information will probably want to enable this option. You can change the example string for whatever you want. This is a security feature.

chroot_local_user=YES
The "chroot_local_user" option is used to control whether local users will be placed in a chroot jail in their home directory after login. It is highly recommended to enable this option for local users FTP access. This is very important if you want to have a secure FTP server where local users cannot access other local users directories. When enabling this option, you have to be sure that users do NOT have shell access on the system, therefore don't forget that when creating a new FTP user, you have to create it with NO shell access. This is a security feature and the most important option of the configuration file.

Step2
Now, set the permission mode of the vsftpd.conf file to be (0600/-rw-------) and owned by the super-user 'root' for security reasons.

- To change the permission mode and ownership of the vsftpd.conf file, use the following commands:

```
[root@deep /]# chmod 600 /etc/vsftpd.conf
[root@deep /]# chown 0.0 /etc/vsftpd.conf
```

/etc/pam.d/ftp: The vsFTPd PAM Support Configuration File

vsFTPd is made to run with the PAM mechanism for password authentication of local FTP users.

Step 1
To be able to use this feature, we must create the /etc/pam.d/ftp file and add the following parameters inside it.

- Create the **ftp** file (touch /etc/pam.d/ftp) and add the following lines:

```
#%PAM-1.0
auth       required     /lib/security/pam_listfile.so item=user
sense=deny file=/etc/ftpusers onerr=succeed
auth       required     /lib/security/pam_stack.so service=system-auth
auth       required     /lib/security/pam_shells.so
account    required     /lib/security/pam_stack.so service=system-auth
session    required     /lib/security/pam_stack.so service=system-auth
```

Step2
Now, set the permission mode of the ftp file to be (0640/-rw-r-----) and owned by the super-user 'root' for security reasons.

- To change the permission mode and ownership of the ftp file, use the commands:

```
[root@deep /]# chmod 640 /etc/pam.d/ftp
[root@deep /]# chown 0.0 /etc/pam.d/ftp
```

/etc/ftpusers: The vsFTPd Access Configuration File

This file is used to define a list of users from which access to the FTP server is always denied. This is a security file where we list all system users that should never get access to the FTP server due to the nature of their UID/GID privilege on the operating system.

Step 1
Please fill free to add to the list below, all users from which you don't want to allow FTP access.

- Create the **ftpusers** file (touch /etc/ftpusers) and add the following lines:

```
root
bin
daemon
sync
mail
nobody
named
rpm
www
amavis
mysql
```

Step 2
Now, set the permission mode of the ftpusers file to be (0600/-rw-------) and owned by the super-user 'root' for security reasons.

- To change the permission mode and ownership of the ftpusers file, use:
```
[root@deep /]# chmod 600 /etc/ftpusers
[root@deep /]# chown 0.0 /etc/ftpusers
```

/etc/logrotate.d/vsftpd: The vsFTPd Log rotation File

The /etc/logrotate.d/vsftpd file allows the FTP server to rotate each week all vsFTPd log files automatically.

- Create the **vsftpd** file (touch /etc/logrotate.d/vsftpd) and add the lines:

```
/var/log/vsftpd.log {
    nocompress
    missingok
}
```

/etc/init.d/vsftpd: The vsFTPd Initialization File

The /etc/init.d/vsftpd script file is responsible for automatically starting and stopping the vsFTPd server on your system. It is important to note that the script will not work if the tcpserver binary available from ucspi-tcp is not also installed.

Refer to the ucspi-tcp chapter in this book if you need more information about ucspi-tcp or want to install and use it with vsFTPd.

The following script is suitable for Linux operating systems that use SystemV. If you Linux system use some other methods like BSD, you'll have to adjust the script bellow to make it work for you.

Step 1
Create the **vsftpd** script file (touch /etc/init.d/vsftpd) and add the following lines:

```
#!/bin/bash

# This shell script takes care of starting and stopping vsFTPd (FTP server).
#
# chkconfig: 345 85 15
# description: vsiFTPd is a Very Secure FTP daemon. \
#              It was written completely from scratch.
#
# processname: vsftpd
# config: /etc/vsftpd.conf

# Source function library.
. /etc/init.d/functions

# Source networking configuration.
test -f /etc/sysconfig/network && . /etc/sysconfig/network

RETVAL=0
```

```
start() {
        echo -n "Starting vsFTPd: "
        tcpserver -c 4096 -DRHl localhost 0 21 /usr/sbin/vsftpd &
        RETVAL=$?
        echo
        [ $RETVAL = 0 ] && touch /var/lock/subsys/vsftpd
        return $RETVAL
}

stop() {
        echo -n "Shutting down vsFTPd: "
        killproc tcpserver
        RETVAL=$?
        echo
        [ $RETVAL = 0 ] && rm -f /var/lock/subsys/vsftpd
        return $RETVAL
}

restart() {
        stop
        start
}

condrestart() {
        [ -e /var/lock/subsys/vsftpd ] && restart
        return 0
}

# See how we were called.
case "$1" in
    start)
        start
        ;;
    stop)
        stop
        ;;
    restart)
        restart
        ;;
    condrestart)
        condrestart
        ;;
    *)
        echo "Usage: vsftpd {start|stop|restart|condrestart}"
        RETVAL=1
esac
exit $RETVAL
```

Step 2
Once the vsftpd script file has been created, it is important to make it executable, change its default permissions, create the necessary links and start it. Making this file executable will allow the system to run it, changing its default permission is to allow only the root user to change this file for security reasons, and creation of the symbolic links will let the process control initialization start the program automatically for you at each system boot.

- To make this script executable and to change its default permissions, use the commands:
```
[root@deep /]# chmod 700 /etc/init.d/vsftpd
[root@deep /]# chown 0.0 /etc/init.d/vsftpd
```

- To create the symbolic rc.d links for vsFTPd, use the following commands:
```
[root@deep /]# chkconfig --add vsftpd
[root@deep /]# chkconfig --level 345 vsftpd on
```

- To start vsFTPd software manually, use the following command:
```
[root@deep /]# /etc/init.d/vsftpd start
Starting vsFTPd:          [OK]
```

Creating an account for FTP client to connect to the FTP server

Once vsFTPd is running on your server, it's time to create an FTP user account on the system to allow FTP client connection to the FTP server. Here are the steps to follow each time you want to add a new FTP user to your FTP server.

Step 1
It's important to give to your, strictly, FTP user NO real shell account on the system. In this manner, if for any reason someone could successfully get out of the FTP chrooted environment; they would not be able to use a shell to gain access via other protocols like telnel, ssh, etc.

First, create the new user for this purpose; this user will be the user allowed to connect to your FTP server. This has to be separate from a regular user account with unlimited access because of how the "chroot" environment works. Chroot makes it appear from the user's perspective as if the level of the file system you've placed it in is the top level.

Here we create a new FTP local user called "gmourani" for this example.

- Use the following command to create a new FTP local user. This step must be done for each additional new local user you allow to access your FTP server on OpenNA Linux.
```
[root@deep /]# useradd -m -d /home/httpd/gmourani -s /bin/false gmourani
[root@deep /]# passwd gmourani
Changing password for user gmourani
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully
```

- Use the following command to create a new FTP local user. This step must be done for each additional new local user you allow to access your FTP server on Red Hat Linux.
```
[root@deep /]# useradd -g users -d /home/httpd/gmourani -s /bin/false
gmourani

[root@deep /]# passwd gmourani
Changing password for user gmourani
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully
```

The useradd command will add the new local user called "gmourani" to our Linux server. The passwd command will set the password for this local user "gmourani".

Contrary to what we have used to create FTP client account with ProFTPD, here the command changes a little bit. We add the option "-d /home/httpd/gmourani" to inform the system to create the FTP home directory for this user under the /home/httpd directory.

With vsFTPd, this is very important, since the program automatically chroot's the specified user into its home directory once logged. If we want to provide web hosting for our users, we need to specify the home directory of the users to be located under the /home/httpd directory because it's a common place when we use Apache to host web site for our users.

Setup an anonymous FTP server

For anonymous FTP server access with vsFTPd, we don't need to create any special directories or even copy binaries or libraries into the anonymous FTP directory to make it work. Anonymous FTP access is really easy to setup with vsFTPd, all you need is to change the default vsftpd.conf file to make it work for anonymous FTP server and create the required anonymous directory on the system.

In our example we'll first give anonymous users access to only get files from the FTP anonymous directory on the FTP server and further down, show you how to setup the configuration file of vsFTPd to allow anonymous users to upload into a specific subdirectory of the FTP anonymous directory.

Step 1
First, we must create the anonymous directory on our server and change its mode permission to allow anonymous FTP access on the server. We decide to create the anonymous directory under the /home directory of the server and call it "ftp". Here are the steps to do it.

- To create the anonymous directory called "ftp" with the correct permission mode on the server, use the following commands:

```
[root@deep /]# mkdir /home/ftp
[root@deep /]# chown -R ftp.ftp /home/ftp/
[root@deep /]# chmod -R 0555 /home/ftp/
```

The above command will create the /home/ftp directory and will change the owner and group of the /home/ftp directory to become the user and group called "ftp". The user and group "ftp" should already exist on your server; remember that we've created it previously. The "chmod" command is important here since vsFTPd does not allow the main anonymous directory to have write access for security reasons; therefore we set its permission mode to (dr-xr-xr-x).

Step2
Once the anonymous directory (/home/ftp) has been created with the correct permissions and modes, it's time to edit our default vsftpd.conf file and do the appropriate changes to make it run as an anonymous FTP server.

Here is what we recommend you to setup. Most options are the same as for the above configuration for local users FTP access. We explain only new options that do not appear under the local users FTP access configuration. The text in bold is what we've added to the configuration file.

- Edit the **vsftpd.conf** file (vi /etc/vsftpd.conf) and set your needs. Below is what we recommend you for anonymous FTP access:

```
anon_root=/home/ftp
dirmessage_enable=YES
xferlog_enable=YES
connect_from_port_20=NO
one_process_model=NO
nopriv_user=ftp
ftpd_banner=OpenNA Linux Anonymous FTP server ready.
```

This tells vsftpd.conf file to set itself up for this particular configuration with:

anon_root=/home/ftp
The "anon_root" option is used to inform the server about the location of the default anonymous directory that vsFTPd will change into after an anonymous login. Remember that we have previously created the /home/ftp directory for this purpose and here we inform vsFTPd about its location on the system.

dirmessage_enable=YES
The "dirmessage_enable" option if set to "YES" will enable any available ASCII text filename (.message) to be displayed to the user the first time they change into a directory (via CWD) per a given session. The file will also be displayed if vsFTPd detects that its last modification time has changed since the previous CWD into a given directory. The file is searched inside all accessible directories of the anonymous /home/ftp directory. You can use as many ASCII text filename (.message) as you want into each directory.

NOTE: Don't forget to restart your FTP server for the changes to take effect.

```
[root@deep /]# /etc/init.d/vsftpd restart
Shutting down vsFTPd:              [OK]
Starting vsFTPd:                   [OK]
```

Allow anonymous users to upload to the FTP server

Once our configuration for an anonymous FTP access is running properly, we can decide to allow anonymous users to upload on the anonymous FTP server subdirectory of our choice. This kind of setting is sometimes required by some administrators. Below are the steps to follow if you want to allow anonymous users to be able to upload into your anonymous FTP server.

Step 1
We have to create a subdirectory inside the existing anonymous directory on our server and change its mode permissions to allow anonymous uploads into this subdirectory of the FTP server. For this example, we decide to name the upload subdirectory "uploads". Here are the steps to do it.

- To create the upload subdirectory called "uploads" with the correct permission modes on the server, use the following commands:
```
[root@deep /]# mkdir /home/ftp/uploads
[root@deep /]# chown -R ftp.ftp /home/ftp/uploads/
```

The above command will create the /home/ftp/uploads subdirectory and will change the owner and group of the /home/ftp/uploads subdirectory to become the user and group called "ftp". The "uploads" subdirectory must have its mode permissions set to (drwxr-xr-x), which is the default for anonymous upload, to work on this subdirectory.

Step2
Once the upload subdirectory (/home/ftp/uploads) has been created with the correct mode permissions on the server, it time to edit our default anonymous vsftpd.conf file and make the appropriated changes to allow upload inside the anonymous FTP server.

Here are the options to add to your default anonymous vsftpd.conf file. Most options are the same as for the anonymous configuration FTP access. We explain only the new options that do not appear under the anonymous FTP access configuration. Text in bold is what we've added to the default anonymous configuration file.

- Edit the **vsftpd.conf** file (vi /etc/vsftpd.conf). Below is what we recommend for anonymous FTP access with upload capability:

```
anon_root=/home/ftp
write_enable=YES
anon_umask=022
anon_upload_enable=YES
chown_uploads=YES
chown_username=ftp
dirmessage_enable=YES
xferlog_enable=YES
connect_from_port_20=NO
one_process_model=NO
nopriv_user=ftp
ftpd_banner=OpenNA Linux Anonymous FTP server ready.
```

This tells the vsftpd.conf file to set itself up for this particular configuration with:

write_enable=YES
The "write_enable" option is used to control whether any FTP commands related to file system change would be allowed on the FTP server. For an anonymous connection with upload capability, where anonymous users are allowed to upload files to a specific subdirectory of the FTP server, we have to say "YES" here because we want to allow anonymous users to upload files on the server.

anon_umask=022
The "anon_umask" option is used to set the default umask value to use for file creation of anonymous users on the FTP server. The value you enter here will represent the permission mode of newly created files by anonymous users on the FTP server. The value of "022" allows new files to be created on the FTP server with mode permissions set to 0644 (-rw-r--r--), which is a safe mode. This is a security feature.

anon_upload_enable=YES
The "anon_upload_enable" option is used to control whether anonymous users will be permitted to upload files on the FTP server. For this to work, the anonymous "ftp" user must have write permission on desired upload locations.

chown_uploads=YES
The "chown_uploads" option is used to control whether all anonymously uploaded files will have the ownership changed to the user specified in the setting "chown_username" below. This is a security feature.

chown_username=ftp
The "chown_username" option is used to specify the name of the user who is given ownership of anonymously uploaded files on the FTP server. In our setting, this name is "ftp", the user under which the FTP server is running. This is a security feature.

NOTE: Don't forget to restart your FTP server for the changes to take effect.

```
[root@deep /]# /etc/init.d/vsftpd restart
Shutting down vsFTPd:               [OK]
Starting vsFTPd:                    [OK]
```

Further documentation

For more details, there are some manual pages about vsFTPd that you could read:

```
$ man vsftpd.conf (5)          - The configuration file for vsFTPd.
$ man vsftpd (8)               - Very Secure FTP Daemon.
```



Part XIV

Hypertext Transfer Protocol

CHAPTER 43

Apache

IN THIS CHAPTER

1. Compiling - Optimizing & Installing `Apache`
2. Configuring `Apache`
3. Running `Apache` with `TLS/SSL` support
4. Running `Apache` in a chroot jail
5. Running `Apache` with users authentication support
6. Caching frequently requested static files
7. Some statistics about `Apache` and Linux

Linux Apache

Abstract

Apache is the most widely used HTTP-server in the world today. It surpasses all free and commercial competitors on the market, and provides a myriad of features; more than the nearest opponent could give you on a UNIX variant. It is also the most used web server for a Linux system.

A web server like Apache, in its simplest function, is software that displays and serves HTML pages hosted on a server to a client browser that understands the HTML code. Mixed with third party modules and programs, it can become powerful software, which will provide strong and useful services to a client browser.

I expect that most of the users that read this book will be especially interested in knowing how to install the Apache web server in the most secure, and optimized, way. In its base install, Apache is no more difficult to install then the other software we have installed on our Linux server. The procedures can become tricky when we want to add some third party modules or programs.

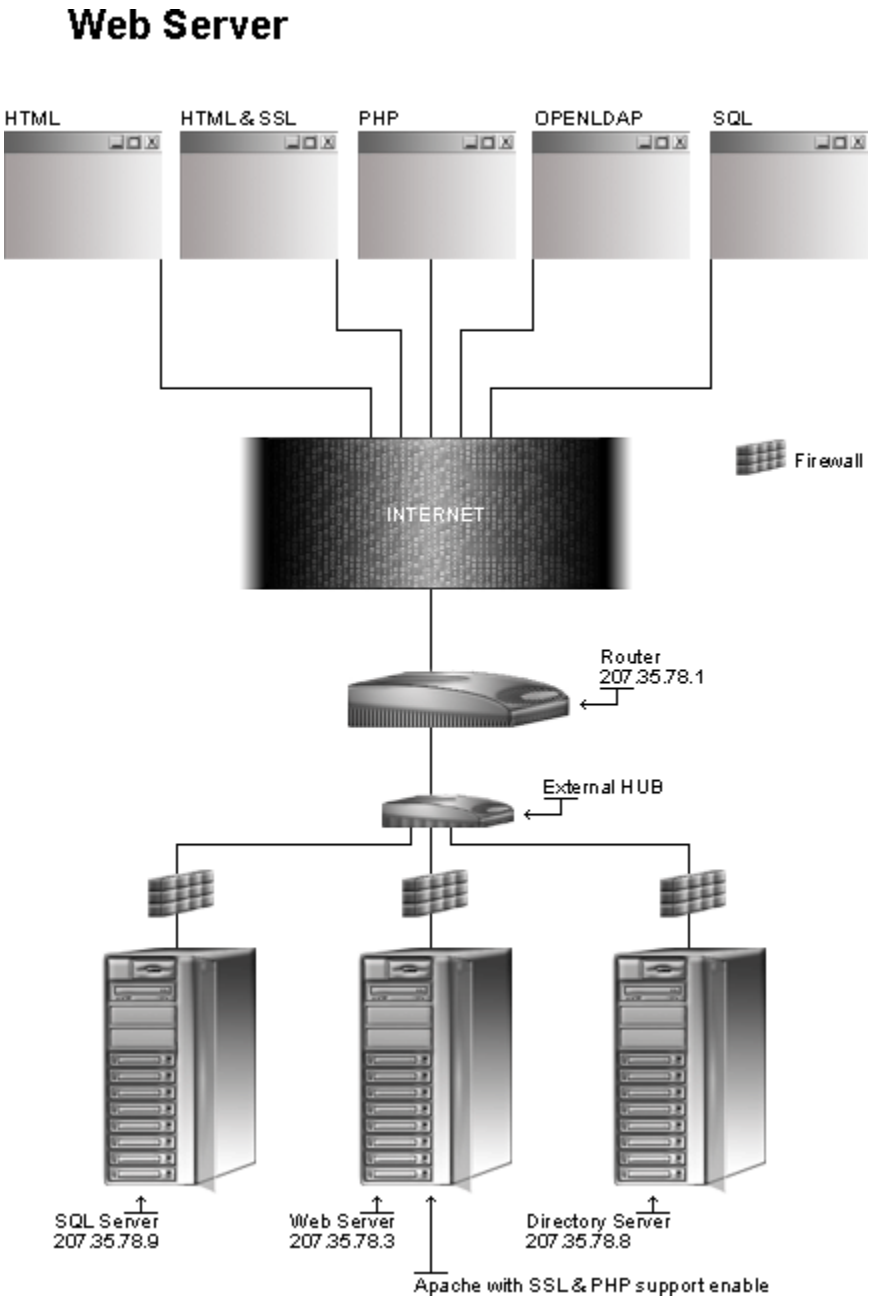
There are a lot of possibilities, variants and options for installing Apache. Therefore, in the following, we provide some step-by-step examples where you can see how to build Apache with other third-party modules and programs like mod_ssl, mod_perl, PHP4, SQL database, etc.

Of course, the building of these programs is optional, and you are free to compile only what you want. In this chapter, we explain and cover some of the basic ways in which you can adjust the configuration to improve the server's performance. Also, for the interested users, we'll provide a procedure to be able to run Apache as a non root-user and in a chrooted environment for optimal security.

After a long period of time and development, the Apache group has finally produced a new generation of its web server. This new web server will become the de-facto standard in the web server world with all of its new features and improvement. At this time, I prefer to inform you that Apache new generation (2.x) is still under development and I consider it as experimental again.

In regard to how I've explained how to install Apache in previous version of this book, you will find here that I've decided to show you how to install it with modules support also know as DSO. This approach is completely different from static build of the software and better now because most of us will compile the software with many external supports like SQL, PHP, IMAP, etc. In this way it is better to have a modularized web server where modules can be loaded as demand for some simple reason; the Apache binary is smaller in size and this provide better performance. In previous setup we compiled everything directly inside the code and test show us that bigger binary is slower than smaller binary.

Therefore if we consider the number of external features that we will provide with the web server as a loadable module compared to the way of compiling these features directly inside the httpd code, we can conclude that Apache will run faster when many features are available as modules instead of begging compiled inside its source code because the resulting binary is smaller to execute for the operating system and heat less memory of the system.


Web Server
HTML
HTML & SSL
PHP
OPENLDAP
SQL
Firewall
INTERNET
Router
207.35.78.1
External HUB
SQL Server
207.35.78.9
Web Server
207.35.78.3
Directory Server
207.35.78.8
Apache with SSL & PHP support enable

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, at personal discretion).
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest Apache version number is 2.0.39

The procedures given in this chapter are likely to work on all Linux platforms, but we have only tested it on OpenNA Linux and Red Hat Linux.

Packages

The following is based on information listed by Apache as of 2002/06/24. Please regularly check http://httpd.apache.org/ for the latest status. We chose to install the required component from a source file because it provides the facility to fine tune the installation.

Source code is available from:
Apache Homepage: http://httpd.apache.org/
Apache FTP Site: 198.3.136.138
You must be sure to download: httpd-2.0.39.tar.gz

Prerequisites

Apache requires that the software below is already installed on your system to be able to compile successfully. If this is not the case, you must install it. Please make sure you have this program installed on your machine before you proceed with this chapter.

- ✓ OpenSSL is required to be able to use Apache with SSL support in your system.
- ✓ expat package is required to be able to use Apache in your system.
- ✓ expat-devel package is required to be able to build Apache in your system.
- ✓ gdbm-devel package is required to be able to build Apache in your system.
- ✓ db3-devel package is required to be able to build Apache in your system.
- ✓ Perl package is required to be able to use Apache in your system.

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all the files installed on the system in the eventuality of an update in the future. To solve the problem, it is a good idea to make a list of files on the system before you install Apache, and then one afterwards, and then compare them using the **diff** utility to find out what files were placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep root]# find /* > Apache1
  ```
- And the following one after you install the software:
  ```
  [root@deep root]# find /* > Apache2
  ```
- Then use the following command to get a list of what changed:
  ```
  [root@deep root]# diff Apache1 Apache2 > Apache-Installed
  ```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. In the example above, we use the /root directory of the system to store all generated list files.

Compiling - Optimizing & Installing Apache

Below are the steps that you must make to configure, compile and optimize the Apache software before installing it onto your system. First off, we install the program as the user "root" so as to avoid permissioning problems.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- This can be done with the following commands:
  ```
  [root@deep /]# cp httpd-version.tar.gz /var/tmp/
  [root@deep /]# cd /var/tmp/
  [root@deep tmp]# tar xzpf httpd-version.tar.gz
  ```

Step 2
In order to check that the version of Apache, which you are, going to install, is an original and unmodified one, please check the supplied signature with the PGP key of Apache available on the Apache website.

To get a PGP key copy of Apache, please point your browser to the following URL: http://www.apache.org/dist/httpd/. For more information about how to use this key for verification, see the GnuPG chapter in this book.

Step 3
Apache cannot run as super-user root; for this reason we must create a special user with no shell privileges on the system for running Apache daemon.

- To create this special Apache user on OpenNA Linux, use the following commands:
```
[root@deep tmp]# groupadd -g 48 www > /dev/null 2>&1 || :
[root@deep tmp]# useradd -c "Web Server" -d /home/httpd -g 48 -s
/bin/false -u 48 www > /dev/null 2>&1 || :
```

- To create this special Apache user on Red Hat Linux, use the following commands:
```
[root@deep tmp]# groupadd -g 48 www > /dev/null 2>&1 || :
[root@deep tmp]# useradd -u 48 -g 48 -s /bin/false -M -r -d /home/httpd
www > /dev/null 2>&1 || :
```

The above command will create a null account, with no password, no valid shell, no files owned-nothing but a UID and a GID for the program. Remember that Apache daemon does not need to have a shell account on the server.

Step 4
Now, edit the **shells** file (vi /etc/shells) and add a non-existent shell name "/bin/false", which is the one we used in the useradd command above.

```
[root@deep tmp]# vi /etc/shells
/bin/bash2
/bin/bash
/bin/sh
/bin/false ← This is our added no-existent shell
```

Step 5
After that, move into the newly created Apache source directory and perform the following steps to configure and optimize Apache for your system.

- To move into the newly created Apache source directory use the command:
```
[root@deep tmp]# cd httpd-2.0.39/
```

Step 6
There are some source files to modify before going in configuration and compilation of the program; the changes allow us to fix some problems and file locations as well as to improve the default number of server processes that we can start of the system.

We begging with the `config.layout` file which relate to location of installed `Apache` files on our system. We must modify it to fit our path environment variable and the way we install `Apache` on the server.

- Edit the **`config.layout`** file (`vi +39 config.layout`) and change the lines:

```
<Layout GNU>
    prefix:        /usr/local
    exec_prefix:   ${prefix}
    bindir:        ${exec_prefix}/bin
    sbindir:       ${exec_prefix}/sbin
    libdir:        ${exec_prefix}/lib
    libexecdir:    ${exec_prefix}/libexec
    mandir:        ${prefix}/man
    sysconfdir:    ${prefix}/etc+
    datadir:       ${prefix}/share+
    installbuilddir: ${datadir}/build
    errordir:      ${datadir}/error
    iconsdir:      ${datadir}/icons
    htdocsdir:     ${datadir}/htdocs
    manualdir:     ${datadir}/manual
    cgidir:        ${datadir}/cgi-bin
    includedir:    ${prefix}/include+
    localstatedir: ${prefix}/var+
    runtimedir:    ${localstatedir}/run
    logfiledir:    ${localstatedir}/log
    proxycachedir: ${localstatedir}/proxy
</Layout>
```

To read:

```
<Layout GNU>
    prefix:        /usr
    exec_prefix:   ${prefix}
    bindir:        ${exec_prefix}/bin
    sbindir:       ${exec_prefix}/sbin
    libdir:        ${exec_prefix}/lib
    libexecdir:    ${libdir}/apache
    mandir:        ${prefix}/share/man
    sysconfdir:    /etc/httpd
    datadir:       /home/httpd
    installbuilddir: ${libexecdir}/build
    errordir:      ${datadir}/error
    iconsdir:      ${datadir}/icons
    htdocsdir:     ${datadir}/html
    manualdir:     ${datadir}/manual
    cgidir:        ${datadir}/cgi-bin
    includedir:    ${prefix}/include/apache
    localstatedir: /var
    runtimedir:    ${localstatedir}/run
    logfiledir:    ${localstatedir}/log/httpd
    proxycachedir: ${localstatedir}/proxy
</Layout>
```

Step 7
For some reason when Apache builds the apxs Perl script, it sometimes ends up getting built without the proper compiler, flags variables and location. We need to solve this problem now before compiling the Apache web server or the generated apxs script file will fail to work.

- Edit the **apxs.in** file (vi +69 support/apxs.in) and change the lines:

```
get_config_vars("$prefix/build/config_vars.mk",\%config_vars);
```

To read:

```
get_config_vars("$prefix/lib/apache/build/config_vars.mk",\%config_vars);
```

- Edit the **apxs.in** file (vi +421 support/apxs.in) and change the lines:

```
push(@cmds, "$prefix/build/libtool $ltflags --mode=compile $CFG_CC
$cflags -I$CFG_INCLUDEDIR $opt -c -o $lo $s && touch $slo");
```

To read:

```
push(@cmds, "$prefix/bin/libtool $ltflags --mode=compile $CFG_CC $cflags
-I$CFG_INCLUDEDIR $opt -c -o $lo $s && touch $slo");
```

- Edit the **apxs.in** file (vi +446 support/apxs.in) and change the lines:

```
push(@cmds, "$prefix/build/libtool $ltflags --mode=link $CFG_CC -o
$dso_file -rpath $CFG_LIBEXECDIR -module -avoid-version $opt $lo");
```

To read:

```
push(@cmds, "$prefix/bin/libtool $ltflags --mode=link $CFG_CC -o
$dso_file -rpath $CFG_LIBEXECDIR -module -avoid-version $opt $lo");
```

- Edit the **apxs.in** file (vi +478 support/apxs.in) and change the lines:

```
push(@cmds, "$prefix/build/instdso.sh SH_LIBTOOL='" .
      "$prefix/build/libtool' $f $CFG_LIBEXECDIR");
```

To read:

```
push(@cmds, "$prefix/lib/apache/build/instdso.sh SH_LIBTOOL='" .
      "$prefix/bin/libtool' $f $CFG_LIBEXECDIR");
```

Step 8
The maximum number of child processes that could be created to serve requests is limited by default to "256" into the source code of Apache. This limit is only valid for the prefork model of the Apache web server. For highly loaded web server, we should increase this limit to "1024" for better performance. This can be done by editing the related source file inside the Apache source directory.

- Edit the **prefork.c** file (vi +118 server/mpm/prefork/prefork.c) and change the following line:

  ```
  #define DEFAULT_SERVER_LIMIT 256
  ```

 To read:

  ```
  #define DEFAULT_SERVER_LIMIT 1024
  ```

Step 9
Once the required modifications have been made into the related source files of Apache, it is time configure and optimize it for our system. As you will see further down, in our compilation of the web server, we disable any experimental modules to keep the software scalable, and disable any unneeded modules to avoid possible security hole and to improve performance.

It is important to note that with new the version of Apache, the server ships with a selection of **M**ulti-**P**rocessing **M**odules (MPMs) which are responsible for binding to network ports on the machine, accepting requests, and dispatching children to handle the requests. In regard to previous version of the software, we have the choice to select with MPM we want to implement with the web server.

We can ONLY choose one and only one type of MPM to compile with Apache, where we choose "prefork" to implements a non-threaded, pre-forking web server that handles requests in a manner similar to Apache 1.3. It's vital to choose this type of MPM now because other are too experimental at this time to be used on production server and choosing something else than "prefork" as the MPM for Apache 2 will certainly break other kind of modules like PHP, Mod_Perl, etc.

- To compile and optimize Apache use the following compilation lines:

```
CFLAGS="-O2 -march=i686 -funroll-loops -fPIC"; export CFLAGS
./configure \
--enable-layout=GNU \
--prefix=/usr \
--exec-prefix=/usr \
--bindir=/usr/bin \
--sbindir=/usr/sbin \
--mandir=/usr/share/man \
--sysconfdir=/etc/httpd \
--includedir=/usr/include/apache \
--libexecdir=/usr/lib/apache \
--datadir=/home/httpd \
--localstatedir=/var \
--enable-access=shared \
--enable-actions=shared \
--enable-alias=shared \
--enable-auth=shared \
--enable-auth-dbm=shared \
--enable-auth-digest=shared \
```

```
--enable-autoindex=shared \
--enable-cern-meta=shared \
--enable-cgi=shared \
--enable-dav=shared \
--enable-dav-fs=shared \
--enable-dir=shared \
--enable-env=shared \
--enable-expires=shared \
--enable-file-cache=shared \
--enable-headers=shared \
--enable-include=shared \
--enable-log-config=shared \
--enable-mime=shared \
--enable-mime-magic=shared \
--enable-negotiation=shared \
--enable-rewrite=shared \
--enable-setenvif=shared \
--enable-speling=shared \
--enable-ssl=shared \
--enable-unique-id=shared \
--enable-usertrack=shared \
--enable-vhost-alias=shared \
--disable-auth-anon \
--disable-charset-lite \
--disable-disk-cache \
--disable-mem-cache \
--disable-cache \
--disable-deflate \
--disable-ext-filter \
--disable-case-filter \
--disable-case-filter-in \
--disable-example \
--disable-proxy \
--disable-proxy-connect \
--disable-proxy-ftp \
--disable-proxy-http \
--disable-status \
--disable-asis \
--disable-info \
--disable-suexec \
--disable-cgid \
--disable-imap \
--disable-userdir \
--with-z \
--with-ssl \
--with-mpm=prefork
```

WARNING: It's important to note that removing all unneeded modules during the configure time of `Apache` will improve the performance of your web server. In our configuration, we've removed the most unused modules both to lower the load operation, and limit the security risks in our `Apache` web server. See your `Apache` documentation for information on each one.

Step 10
At this stage the program is ready to be built and installed. We build Apache with the 'make' command and produce a list of files on the system before we install the software, and one afterwards, then compare them using the **diff** utility to find out what files were placed where and finally install Apache.

```
[root@deep httpd-2.0.39]# make
[root@deep httpd-2.0.39]# cd
[root@deep root]# find /* > Apache1
[root@deep root]# cd /var/tmp/httpd-2.0.39/
[root@deep httpd-2.0.39]# make install
[root@deep httpd-2.0.39]# strip /usr/sbin/httpd
[root@deep httpd-2.0.39]# chmod 0511 /usr/sbin/httpd
[root@deep httpd-2.0.39]# strip -R .comment /usr/lib/apache/*.so
[root@deep httpd-2.0.39]# cd
[root@deep root]# find /* > Apache2
[root@deep root]# diff Apache1 Apache2 > Apache-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

Step 11
Once the compilation, optimization and installation of the software has completed, we can free up some disk space by deleting the program tar archive and the related source directory, since they are no longer needed.

- To delete Apache and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf httpd-version/
[root@deep tmp]# rm -f httpd-version.tar.gz
```

The rm command as used above will remove all the source files we have used to compile and install Apache. It will also remove the Apache compressed archive from the /var/tmp directory.

Configuring Apache

After Apache has been built and installed successfully on your system, the next step is to configure and customize its configuration files to fit your needs.

- ✓ /etc/httpd/httpd.conf: (The Apache Configuration File)
- ✓ /etc/logrotate.d/httpd: (The Apache Log Rotation File)
- ✓ /etc/sysconfig/httpd: (The Apache System Configuration File)
- ✓ /etc/init.d/httpd: (The Apache Initialization File)

/etc/httpd/httpd.conf: The Apache Configuration File

The httpd.conf file is the main configuration file for the Apache web server. A lot options exist, and it's important to read the documentation that comes with Apache for more information on different settings and parameters.

The following configuration is a full and secure working configuration file for Apache. Also, it's important to note that I only comment parameters that relate to security and optimization, and leave all the others to your own research. Text in bold is the parts of the configuration file that must be customized and adjusted to satisfy your needs.

- Edit the **`httpd.conf`** file (`vi /etc/httpd/httpd.conf`) and set your needs:

```
### Section 1: Global Environment
#
ServerRoot "/etc/httpd"
LockFile /var/log/httpd/accept.lock
PidFile /var/run/httpd.pid

Timeout 120
KeepAlive On
MaxKeepAliveRequests 0
KeepAliveTimeout 10

StartServers 5
MaxClients 512
ServerLimit 1024
MinSpareServers 32
MaxSpareServers 64
MaxRequestsPerChild 0

Listen 127.0.0.1:80
Listen 127.0.0.1:443

LoadModule access_module        lib/apache/mod_access.so
LoadModule alias_module         lib/apache/mod_alias.so
LoadModule autoindex_module     lib/apache/mod_autoindex.so
LoadModule cgi_module           lib/apache/mod_cgi.so
LoadModule dir_module           lib/apache/mod_dir.so
LoadModule env_module           lib/apache/mod_env.so
LoadModule include_module       lib/apache/mod_include.so
LoadModule log_config_module    lib/apache/mod_log_config.so
LoadModule mime_module          lib/apache/mod_mime.so
LoadModule mime_magic_module    lib/apache/mod_mime_magic.so
LoadModule rewrite_module       lib/apache/mod_rewrite.so
LoadModule setenvif_module      lib/apache/mod_setenvif.so
#LoadModule php4_module          lib/apache/libphp4.so
#LoadModule ssl_module           lib/apache/mod_ssl.so
#LoadModule perl_module          lib/apache/mod_perl.so
#LoadModule actions_module      lib/apache/mod_actions.so
#LoadModule auth_module         lib/apache/mod_auth.so
#LoadModule auth_dbm_module     lib/apache/mod_auth_dbm.so
#LoadModule auth_digest_module  lib/apache/mod_auth_digest.so
#LoadModule cern_meta_module    lib/apache/mod_cern_meta.so
#LoadModule dav_module          lib/apache/mod_dav.so
#LoadModule dav_fs_module       lib/apache/mod_dav_fs.so
#LoadModule expires_module      lib/apache/mod_expires.so
#LoadModule file_cache_module   lib/apache/mod_file_cache.so
#LoadModule headers_module      lib/apache/mod_headers.so
#LoadModule negotiation_module  lib/apache/mod_negotiation.so
#LoadModule speling_module      lib/apache/mod_speling.so
#LoadModule unique_id_module    lib/apache/mod_unique_id.so
#LoadModule usertrack_module    lib/apache/mod_usertrack.so
#LoadModule vhost_alias_module  lib/apache/mod_vhost_alias.so


### Section 2: 'Main' server configuration
#
User www
Group www

ServerAdmin root@localhost
ServerName localhost
```

```
UseCanonicalName Off

DocumentRoot "/home/httpd/html"
<Directory />
    Options None
    AllowOverride None
    Order deny,allow
    Deny from all
</Directory>

<Files .pl>
    Options None
    AllowOverride None
    Order deny,allow
    Deny from all
</Files>

<IfModule mod_file_cache.c>
<IfModule mod_include.c>
    Include /etc/httpd/mmap.conf
</IfModule>
</IfModule>

<IfModule mod_dir.c>
    DirectoryIndex index.htm index.html index.php default.php index.php3
</IfModule>

<IfModule mod_mime.c>
    TypesConfig /etc/httpd/mime.types
    AddEncoding x-compress Z
    AddEncoding x-gzip gz tgz
    AddType application/x-tar .tgz
    AddType application/x-httpd-php .php
    AddType application/x-httpd-php .php3
    AddType application/x-httpd-php-source .phps
</IfModule>

DefaultType text/plain

<IfModule mod_mime_magic.c>
    MIMEMagicFile /etc/httpd/magic
</IfModule>

HostnameLookups Off

ErrorLog /var/log/httpd/error_log
LogLevel warn
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\""
combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
CustomLog /var/log/httpd/access_log combined

ServerTokens Prod
ServerSignature Off

<IfModule mod_alias.c>
Alias /icons/ "/home/httpd/icons/"
<Directory "/home/httpd/icons">
    Options None
    AllowOverride None
    Order allow,deny
```

```
    Allow from all
</Directory>

ScriptAlias /cgi-bin/  "/home/httpd/cgi-bin/"
 <Directory "/home/httpd/cgi-bin">
    Options None
    AllowOverride None
    Order allow,deny
    Allow from all
 </Directory>
</IfModule>

<IfModule mod_autoindex.c>
    IndexOptions FancyIndexing
    AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip
    AddIconByType (TXT,/icons/text.gif) text/*
    AddIconByType (IMG,/icons/image2.gif) image/*
    AddIconByType (SND,/icons/sound2.gif) audio/*
    AddIconByType (VID,/icons/movie.gif) video/*
    AddIcon /icons/binary.gif .bin .exe
    AddIcon /icons/binhex.gif .hqx
    AddIcon /icons/tar.gif .tar
    AddIcon /icons/world2.gif .wrl .wrl.gz .vrml .vrm .iv
    AddIcon /icons/compressed.gif .Z .z .tgz .gz .zip
    AddIcon /icons/a.gif .ps .ai .eps
    AddIcon /icons/layout.gif .html .shtml .htm .pdf
    AddIcon /icons/text.gif .txt
    AddIcon /icons/c.gif .c
    AddIcon /icons/p.gif .pl .py
    AddIcon /icons/f.gif .for
    AddIcon /icons/dvi.gif .dvi
    AddIcon /icons/uuencoded.gif .uu
    AddIcon /icons/script.gif .conf .sh .shar .csh .ksh .tcl
    AddIcon /icons/tex.gif .tex
    AddIcon /icons/bomb.gif core
    AddIcon /icons/back.gif ..
    AddIcon /icons/hand.right.gif README
    AddIcon /icons/folder.gif ^^DIRECTORY^^
    AddIcon /icons/blank.gif ^^BLANKICON^^
    DefaultIcon /icons/unknown.gif
    ReadmeName README.html
    HeaderName HEADER.html
    IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t
</IfModule>

ErrorDocument 400 "Server could not understand this request."
ErrorDocument 401 "Server could not verify your access authorization."
ErrorDocument 403 "Access Forbidden -- Go away."
ErrorDocument 404 "/error.htm"
ErrorDocument 405 "Method not allowed for the requested URL."
ErrorDocument 408 "Server closed the network connection."
ErrorDocument 410 "Requested URL no longer available."
ErrorDocument 411 "Requested method requires a valid header."
ErrorDocument 412 "Precondition request failed positive evaluation."
ErrorDocument 413 "Method not allowed for the data transmitted."
ErrorDocument 414 "Requested URL exceeds the capacity limit."
ErrorDocument 415 "Server temporarily unavailable."
ErrorDocument 500 "Server encountered an internal error."
ErrorDocument 501 "Server does not support the action requested."
ErrorDocument 502 "Proxy server received an invalid response."
ErrorDocument 503 "Server temporarily unavailable."
ErrorDocument 506 "Access not possible."
```

```
<IfModule mod_setenvif.c>
BrowserMatch "Mozilla/2" nokeepalive
BrowserMatch "MSIE 4\.0b2;" nokeepalive downgrade-1.0 force-response-1.0
BrowserMatch "RealPlayer 4\.0" force-response-1.0
BrowserMatch "Java/1\.0" force-response-1.0
BrowserMatch "JDK/1\.0" force-response-1.0
BrowserMatch "Microsoft Data Access Internet Publishing Provider"
redirect-carefully
BrowserMatch "^WebDrive" redirect-carefully
</IfModule>


### Section 3: Virtual Hosts
#
NameVirtualHost 127.0.0.1:80

<VirtualHost 127.0.0.1:80>
ServerAdmin root@localhost
ServerName localhost
DocumentRoot "/home/httpd/html"

<Directory "/home/httpd/html">
    Options Indexes MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>

ErrorLog /var/log/httpd/error_log
TransferLog /var/log/httpd/access_log
</VirtualHost>


## SSL Global Context
#
<IfModule mod_ssl.c>
AddType application/x-x509-ca-cert .crt
AddType application/x-pkcs7-crl    .crl

SSLPassPhraseDialog     builtin
SSLSessionCache         none
SSLSessionCacheTimeout  300
SSLMutex                sem
SSLRandomSeed startup   file:/dev/urandom 1024
SSLRandomSeed connect   file:/dev/urandom 1024

## SSL Virtual Host Context
#
NameVirtualHost 127.0.0.1:443

<VirtualHost 127.0.0.1:443>
ServerAdmin root@localhost
ServerName localhost
DocumentRoot "/home/httpd/html"

<Directory "/home/httpd/html">
    Options Indexes MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>

ErrorLog /var/log/httpd/error_log
```

```
TransferLog /var/log/httpd/access_log

SSLEngine on

SSLCipherSuite ALL:!ADH:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:+eNULL
SSLCertificateFile      /usr/share/ssl/certs/www.crt
SSLCertificateKeyFile   /usr/share/ssl/private/www.key
SSLVerifyClient none
SSLVerifyDepth  10

SetEnvIf User-Agent ".*MSIE.*" \
         nokeepalive ssl-unclean-shutdown \
         downgrade-1.0 force-response-1.0

CustomLog /var/log/httpd/ssl_request_log \
          "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"

</VirtualHost>
</IfModule>
```

This tells the `httpd.conf` file to set itself up for this particular configuration setup with:

`ServerRoot "/etc/httpd"`
This directive "`ServerRoot`" is used to define the directory in which the configuration file of the `Apache` server lives. It allows `Apache` to know where it can find its configuration file when it starts. In our setup, this file is located under `/etc/httpd` directory and it's called `httpd.conf`.

`Timeout 120`
This directive "`Timeout`" is used to define the amount of time `Apache` will wait for a `GET`, `POST`, `PUT` request and `ACKs` on transmissions before automatically disconnect when idle time exceeds this value. In our configuration, we set this value to "`120`" to improve performance in heavily loaded servers. It is recommended to set this value lower if your clients have low latencies. Some time, setting this directive to a low value may pause problem, this highly depend of your network and server setup. The best is to experiment with different values to find the one that fit your need. This is a performance feature.

`KeepAlive On`
This directive "`KeepAlive`" if set to "`On`", enables persistent connections on the web server. For better performance, it's recommended to set this option to "`On`" and allow more than one request per connection. In the original `HTTP` specification, every `HTTP` request had to establish a separate connection to the server. To reduce the overhead of frequent connects, the keep-alive header was developed. Keep-alives tells the server to reuse the same socket connection for multiple `HTTP` requests. This is a performance feature.

`MaxKeepAliveRequests 0`
This directive "`MaxKeepAliveRequests`" is used to define the number of requests allowed per connection when the `KeepAlive` option above is set to "`On`". Socket connections will be terminated when the number of requests set by the "`MaxKeepAliveRequests`" directive is reached. When the value of this option is set to "`0`" then unlimited requests are allowed on the server. For server performance, it's recommended to allow unlimited requests. This is a performance feature.

KeepAliveTimeout 10
This directive "KeepAliveTimeout" is used to define how much time, in seconds, Apache will wait for a subsequent request before closing the connection. Once a request has been received, the timeout value specified by the "Timeout" directive applies. The value of "10" seconds is a good average for server performance. This value should be kept low as the socket will be idle for extended periods otherwise. This is a performance feature.

StartServers 5
This directive "StartServers" is used to define the number of child server processes that will be created by Apache on start-up. As the number of processes with Apache 2.x is dynamically controlled depending on the load, there is usually little reason to adjust this parameter now. In our configuration, we use the default value of "5". This is a performance feature.

MaxClients 512
This directive "MaxClients" is used to define the limit on the number of child processes that will be created to serve requests. The default means that up to 512 HTTP requests can be handled concurrently. Any further connection requests are queued. This is an important tuning parameter regarding the performance of the Apache web server. For high load operation, a value of "512" is recommended by various benchmarks on the Internet. For standard use, you can set the value to "256". This is a performance feature.

ServerLimit 1024
This directive "ServerLimit" is used to define the maximum configured value for the "MaxClients" directive for the lifetime of the Apache process. It is important to note that any attempts to change this directive during a restart will be ignored, but the "MaxClients" directive can be modified during a restart of the server. This is another important tuning parameter directly associated with the "MaxClients" directive regarding the performance of the Apache web server. For high load operation, a value of "1024" is highly recommended by various benchmarks on the Internet. For standard use, you can set the value to "256". This is a performance feature.

WARNING: Special care must be taken when using this directive. If "ServerLimit" is set to a value much higher than necessary, extra, unused shared memory will be allocated. If both "ServerLimit" and "MaxClients" are set to values higher than the system can handle, Apache may not start or the system may become unstable.

MinSpareServers 32
This directive "MinSpareServers" is used to define the minimum number of idle child server processes that should be created. An idle process is one which is not handling a request. If there are fewer than "MinSpareServers" idle, then the parent process creates new children at a maximum rate of 1 per second. This is a performance feature.

MaxSpareServers 64
This directive "MaxSpareServers" is used to define the maximum number of idle child server processes that should be created. If there are more than "MaxSpareServers" idle child processes, then the parent process will kill off the excess processes and these extra processes will be terminated. This is a performance feature.

```
MaxRequestsPerChild 0
```
This option "MaxRequestsPerChild" is used to define the number of requests that an individual child server process will handle. In our configuration, we set the value of this directive to "0" to get the maximum performance and scalability for the server. This is an important tuning parameter regarding the performance of the Apache web server again.

```
Listen 1.2.3.4:80
Listen 1.2.3.4:443
```
This directive "Listen" is used to inform the web server to accept incoming requests on the specified port or address-and-port combination. In our example, we define IP address and port number of our web server on the system. Port number "80" is the standard port for HTTP request and "443" is the standard port number for HTTPS request. In this way, we have both ports and IP addresses configured into our configuration file.

```
User www
```
This directive "User" is used to define the UID that Apache daemon will run as. It's important to create a new user that has minimal access to the system, and functions just for the purpose of running the web server daemon. Using a different UID that already exists on the system (i.e. nobody) can allow your services to access each other's resources. In our example, we use the Apache user we have created previously which is called "www".

```
Group www
```
This directive "Group" is used to define the GID the Apache daemon will run as. It's important to create a new group that has minimal access to the system and functions just for the purpose of running the web server daemon. In our example, we use the Apache group we have created previously which is called "www".

```
ServerAdmin root@localhost
```
This directive "ServerAdmin" is used to define the e-mail address that the server includes in any error messages it returns to the client. Don't forget to change the above value to your real email address.

```
ServerName localhost
```
This directive "ServerName" is used to define the hostname that the server uses to identify itself. If your web server is accessible through www.domain.com, then the value of this directive will be www.domain.com. Don't forget to change the above value for your real FQDN.

```
DocumentRoot "/home/httpd/html"
<Directory />
    Options None
    AllowOverride None
    Order deny,allow
    Deny from all
</Directory>
```
This block of directives allows running a really tight ship by stopping users overriding system wide settings. This is because the default Apache access for <Directory /> is Allow from All, and this means that it will serve any file mapped from an URL. For this reason it is highly recommended that you change this block such as the one we have configured and then override this for directories you want accessible. This is a security feature.

```
<IfModule mod_dir.c>
   DirectoryIndex index.htm index.html index.php default.php index.php3
</IfModule>
```

This directive "DirectoryIndex" is used to define the files to use by Apache as a pre-written HTML directory index. In other words, if Apache can't find the default index page to display, it'll try the next entry in this parameter, if available. To improve performance of the web server it's recommended to list the most used default index pages of your web site first and not to include too much. This is a performance feature.

```
HostnameLookups Off
```

This directive "HostnameLookups" if set to "Off", specifies to disable DNS lookups. It's recommended to set this option to "Off" in order to avoid latency to every request, to save the network traffic time, and to improve the performance of your Apache web server. This is a performance feature.

```
ServerTokens Prod
```

This directive "ServerTokens" is used to controls whether server response header field which is sent back to clients includes a description of the generic OS-type of the server as well as information about compiled-in modules. For security reason, I recommend you to limit the number of information send by the web server to the external as much as possible. This is done by setting the value of this directive to "Prod", which means that only the name of the web server (Apache) will be displayed as the information. This is good to avoid version detection with Apache. This is a security feature.

NOTE: If your httpd.conf file contains many <VirtualHost> sections that are substantially the same, then I recommend you to read the Apache "Dynamically configured mass virtual hosting" document, which describes how to efficiently serve an arbitrary number of virtual hosts. This is an online documentation, which can be retrieved from the Apache website at the following URL: http://httpd.apache.org/docs-2.0/vhosts/.

/etc/logrotate.d/httpd: The Apache Log rotation File

The /etc/logrotate.d/httpd file allows the web server to rotate each week all Apache log files automatically.

Step1
Here we'll configure the /etc/logrotate.d/httpd file to rotate each week its log files.

- Create the **httpd** file (touch /etc/logrotate.d/httpd) and add the lines:

```
/var/log/httpd/access_log {
    missingok
    postrotate
        /usr/bin/killall -HUP httpd
    endscript
}

/var/log/httpd/error_log {
    missingok
    postrotate
        /usr/bin/killall -HUP httpd
    endscript
}
```

```
/var/log/httpd/ssl_request_log {
    missingok
    postrotate
        /usr/bin/killall -HUP httpd
    endscript
}
```

NOTE: Lines to automatically rotate the log file called `ssl request log` is included in this file. If you intend to run `Apache` without `SSL` support, you must remove the above lines related to `SSL`.

Step2
Now, set the permission mode of the `httpd` file to be (`0640/-rw-r-----`) and owned by the super-user '`root`' for security reason.

- To change the permission mode and ownership of the `httpd` file, use the commands:
```
[root@deep /]# chmod 640 /etc/logrotate.d/httpd
[root@deep /]# chown 0.0 /etc/logrotate.d/httpd
```

`/etc/sysconfig/httpd`: The `Apache` System Configuration File

The `/etc/sysconfig/httpd` file is used to specify `Apache` system configuration information, such as if additional options are required to be passed to `httpd` daemon at startup.

- Create the **`httpd`** file (`touch /etc/sysconfig/httpd`) and add the following lines:

```
# Uncomment the following line to enable SSL support with Apache.
# Certificate should be already configured into httpd.conf file.
#
#OPTIONS="-DSSL"
```

The "`OPTIONS`" parameter is used to start `Apache` with `SSL` support. If you want to run you web server with `SSL` support, then you have to uncomment this line and add the required certificate to the appropriated directory. This is all you need to do since the initialization file of `Apache` will take care of everything else for you. For now, this line must be commented out since we'll see later in this chapter how to run `Apache` with `SSL` support.

`/etc/init.d/httpd`: The `Apache` Initialization File

The `/etc/init.d/httpd` script file is responsible to automatically start and stop the `Apache` server on your Linux system. Loading the `httpd` daemon as a standalone daemon will eliminate load time and will even reduce swapping since non-library code will be shared.

Please note that the following script is suitable for Linux operating systems that use `System V`. If you Linux system use some other methods like `BSD`, you'll have to adjust the script bellow to make it work for you.

Step 1
Create the **`httpd`** script file (`touch /etc/init.d/httpd`) and add the following lines:

```
#!/bin/bash

# This shell script takes care of starting and stopping Apache.
```

```
#
# chkconfig: 345 85 15
# description: Apache is a World Wide Web server. It is used to serve \
#              HTML files and CGI.
#
# processname: httpd
# config: /etc/httpd/httpd.conf
# pidfile: /var/run/httpd.pid

# Source function library.
. /etc/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Source for additional options if we have them.
if [ -f /etc/sysconfig/httpd ] ; then
        . /etc/sysconfig/httpd
fi

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

# If Apache is not available stop now.
[ -f /usr/sbin/httpd ] || exit 0

# Path to the Apache binary.
httpd=/usr/sbin/httpd

RETVAL=0
prog="Apache"

start() {
        echo -n $"Starting $prog: "
        daemon $httpd $OPTIONS
        RETVAL=$?
        echo
        [ $RETVAL = 0 ] && touch /var/lock/subsys/httpd
        return $RETVAL
}

stop() {
        echo -n $"Shutting down $prog: "
        kill -TERM `cat /var/run/httpd.pid`
        RETVAL=$?
        echo "  [ OK ]"
        [ $RETVAL = 0 ] && rm -f /var/lock/subsys/httpd
        return $RETVAL
}

# See how we were called.
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  status)
        status $httpd
        RETVAL=$?
        ;;
  restart)
```

```
        kill -USR1 `cat /var/run/httpd.pid`
        RETVAL=$?
        ;;
  condrestart)
        if [ -f /var/lock/subsys/httpd ]; then
        kill -USR1 `cat /var/run/httpd.pid`
        RETVAL=$?
        fi
        ;;
  *)
        echo $"Usage: $0 {start|stop|status|restart|condrestart}"
        exit 1
esac
exit $RETVAL
```

Step 2
Once the `httpd` script file has been created, it is important to make it executable, change its default permissions, create the necessary links and start it. Making this file executable will allow the system to run it, changing its default permission is to allow only the root user to change this file for security reason, and creation of the symbolic links will let the process control initialization of Linux which is in charge of starting all the normal and authorized processes that need to run at boot time on your system to start the program automatically for you at each reboot.

- To make this script executable and to change its default permissions, use the command:
```
[root@deep /]# chmod 700 /etc/init.d/httpd
[root@deep /]# chown 0.0 /etc/init.d/httpd
```

- To create the symbolic `rc.d` links for `Apache`, use the following command:
```
[root@deep /]# chkconfig --add httpd
[root@deep /]# chkconfig --level 345 httpd on
```

- To start `Apache` software manually, use the following command:
```
[root@deep /]# /etc/init.d/httpd start
Starting Apache:          [OK]
```

Running `Apache` with `TLS/SSL` support

This section applies only if you want to run `Apache` through `SSL` connection. With the new release of `Apache`, we don't need anymore to use external program like `mod_ssl` to make it work with `SSL` support.

The new generation of `Apache` software comes with its own `SSL` module which is compiled and installed with the software. All we need to do is to enable the `SSL` module as we have already done with our configuration of the web server and create the required certificate to make it work. Below I show you how to set up a certificate to use with `Apache`.

Step 1
First you have to know the **F**ully **Q**ualified **D**omain **N**ame (`FQDN`) of the `Apache` web server for which you want to request a certificate. When you want to access your web server through `www.domain.com` then the `FQDN` of your `Apache` server is `www.domain.com`.

Step 2
Second, select five large and relatively random files from your hard drive (compressed log files are a good start) and put them under your /usr/share/ssl directory. These will act as your random seed enhancers. We refer to them as random1: random2:...: random5 below.

- To select five random files and put them under /usr/share/ssl, use the commands:

```
[root@deep /]# cp /var/log/boot.log /usr/share/ssl/random1
[root@deep /]# cp /var/log/cron /usr/share/ssl/random2
[root@deep /]# cp /var/log/dmesg /usr/share/ssl/random3
[root@deep /]# cp /var/log/messages /usr/share/ssl/random4
[root@deep /]# cp /var/log/secure /usr/share/ssl/random5
```

Step 3
Third, create the RSA private key **not protected with a pass-phrase** for the Apache server. The command below will generate 1024 bit RSA Private Key and stores it in the file www.key.

- To generate the Key, use the following commands:

```
[root@deep /]# cd /usr/share/ssl/
[root@deep ssl]# openssl genrsa -rand
random1:random2:random3:random4:random5 -out www.key 1024
123600 semi-random bytes loaded
Generating RSA private key, 1024 bit long modulus
.....................+++++
.....+++++
e is 65537 (0x10001)
```

WARNING: Please backup your **www.key** file. A good choice is to backup this information onto a diskette or other removable media.

Step 4
Finally, generate a **C**ertificate **S**igning **R**equest (CSR) with the server RSA private key. The command below will prompt you for the X.509 attributes of your certificate. Remember to give the name www.domain.com when prompted for '**Common Name**'. Do not enter your personal name here. We are requesting a certificate for a web server, so the **Common Name** has to match the FQDN of your website.

- To generate the CSR, use the following command:

```
[root@deep ssl]# openssl req -new -key www.key -out www.csr
Using configuration from /usr/share/ssl/openssl.cnf
Enter PEM pass phrase:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [CA]:
State or Province Name (full name) [Quebec]:
Locality Name (eg, city) [Montreal]:
Organization Name (eg, company) [OpenNA, Inc.]:
Organizational Unit Name (eg, section) [OpenNA, Inc. Web Server]:
```

```
Common Name (eg, YOUR name) [www.openna.com]:
Email Address [noc@openna.com]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:.
An optional company name []:.
```

WARNING: Make sure you enter the FQDN (**F**ully **Q**ualified **D**omain **N**ame) of the server when OpenSSL prompts you for the "**Common Name**" (i.e. when you generate a CSR for a web server which will be later accessed via www.domain.com, enter www.domain.com here).

After generation of your **C**ertificate **S**igning **R**equest (CSR), you could send this certificate to a commercial **C**ertifying **A**uthority (CA) like Thawte or Verisign for signing. You usually have to post the CSR into a web form, pay for the signing, await the signed Certificate and store it into an www.crt file. The result is then a real certificate, which can be used for Apache.

Step 5
You are not obligated to send your **C**ertificate **S**igning **R**equest (CSR) to a commercial **C**ertifying **A**uthority (CA) for signing. In some cases and with Apache you can become your own **C**ertifying **A**uthority (CA) and sign your certificate by yourself. In the step below, I assume that your CA keys pair, which is required for signing certificate by yourself, already exists on the server, if this is not the case, please refer to the chapter related to OpenSSL in this book for more information about how to create your CA keys pair and become your own **C**ertifying **A**uthority (CA).

- To sign server CSR's in order to create real SSL certificates, use the following command:

```
[root@deep ssl]# /usr/share/ssl/misc/sign www.csr
CA signing: www.csr -> www.crt:
Using configuration from ca.config
Enter PEM pass phrase:
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName           :PRINTABLE:'CA'
stateOrProvinceName   :PRINTABLE:'Quebec'
localityName          :PRINTABLE:'Montreal'
organizationName      :PRINTABLE:'OpenNA, Inc.'
organizationalUnitName:PRINTABLE:'OpenNA, Inc. Web Server'
commonName            :PRINTABLE:'www.openna.com'
emailAddress          :IA5STRING:'noc@openna.com'
Certificate is to be certified until Mar 15 07:15:45 2002 GMT (365 days)
Sign the certificate? [y/n]: y


1 out of 1 certificate requests certified, commit? [y/n] y
Write out database with 1 new entries
Data Base Updated
CA verifying: www.crt <-> CA cert
www.crt: OK
```

This signs the CSR and results in a **www.crt** file.

Step 6
Now, we must place the certificates files (www.key and www.crt) to the appropriate directories and change their default permission modes to be (0400/-r--------), owned by the user called 'www' for Apache to be able to find and use them when it will start its daemon.

- To place the certificates into the appropriate directory, use the following commands:

```
[root@deep ssl]# mv www.key private/
[root@deep ssl]# mv www.crt certs/
[root@deep ssl]# chmod 400 private/www.key
[root@deep ssl]# chmod 400 certs/www.crt
[root@deep ssl]# chown www.www private/www.key
[root@deep ssl]# chown www.www certs/www.crt
[root@deep ssl]# rm -f www.csr
```

First we move the www.key file to the private directory and the www.crt file to the certs directory. After that we change the permission mode and ownership of both certificates to be only readable and owned by the Apache user called 'www' for security reasons. Finally we remove the www.csr file from our system since it is no longer needed.

Step 7
To allow TLS/SSL-enabled connections with Apache, we must start its daemon with SSL support. This is possible by editing the /etc/sysconfig/httpd file and uncomments the related line as follow.

- Edit the **httpd** file (vi /etc/sysconfig/httpd), and change the line:

```
#OPTIONS="-DSSL"
```

To read:

```
OPTIONS="-DSSL"
```

Step 8
For Apache to know about the certificate files, we have to edit its httpd.conf file and inform it about the location of the certificate files to use for the encrypted connection. In our configuration of the web server, we have already defined the location of the certificates. Therefore we don't need to do it again but I prefer to show you how the configuration lines should look inside your httpd.conf file.

```
SSLCertificateFile      /usr/share/ssl/certs/www.crt
SSLCertificateKeyFile   /usr/share/ssl/private/www.key
```

In this example, www.crt is our web server **C**ertificate **S**igning **R**equest public key, and www.key is our web server RSA private key. Don't forget to configure the virtual section of httpd.conf to make the web server work and find the certificates for the corresponding site. You must configure the virtual section of the SSL part even if you don't use virtual hosting on your web server; this is a requirement for Apache to work with SSL support. Read the Apache documentation if you have some question about the way to do it.

Step 9
As you supposed to know now, SSL capability is available with Apache via module. We have to activate the module for the web server to run with SSL support. This is possible by uncomment the line related to the SSL module inside the httpd.conf file.

- Edit the **httpd.conf** file (vi /etc/httpd/httpd.conf), and change the line:

  ```
  #LoadModule ssl_module           lib/apache/mod_ssl.so
  ```

 To read:

  ```
  LoadModule ssl_module           lib/apache/mod_ssl.so
  ```

Step 10
The Apache TLS/SSL-enabled connections run by default on port 443. To allow external traffic through this port (443), we must enable rules into our firewall script file for the web server to accept external secure connections on the system.

Step 11
Finally, we must restart our Apache server for the changes to take effect.

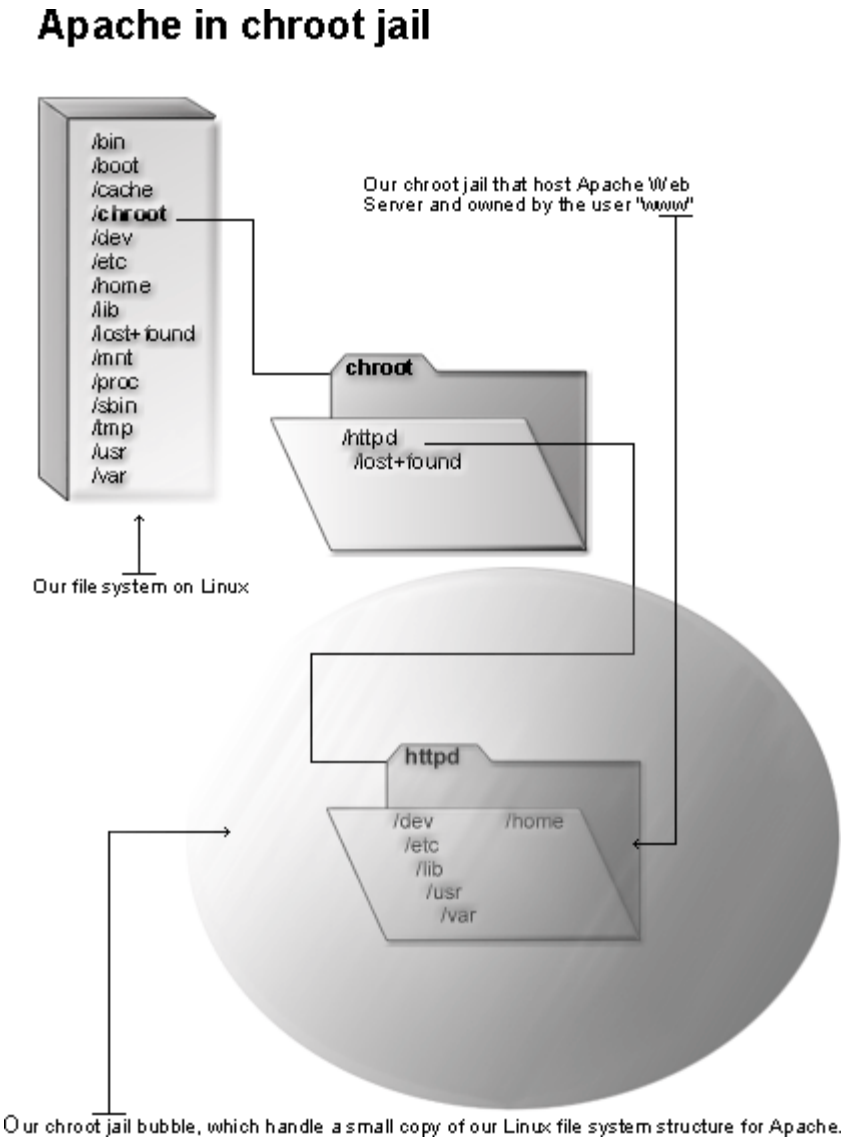
- To restart Apache use the following command:

  ```
  [root@deep /]# /etc/init.d/httpd restart
  Stopping Apache:            [OK]
  Starting Apache:            [OK]
  ```

Running Apache in a chroot jail

This part focuses on preventing Apache from being used as a point of break-in to the system hosting it. The main benefit of a chroot jail is that the jail will limit the portion of the file system the daemon can see to the root directory of the jail.

Additionally, since the jail only needs to support Apache, the programs available in the jail can be extremely limited. Most importantly, there is no need for setuid-root programs, which can be used to gain root access and break out of the jail. By running Apache in a chroot environment you can improve the security significantly in a Unix environment.



Chrooting `Apache` is not an easy task and has a tendency to break things. Before we embark on this, we need to first decide whether it is beneficial for you to do so. Some pros and cons are, but most certainly not limited to, the following:

Pros:

- ✓ If `Apache` is ever compromised, the attacker will not have access to the entire Linux OS.
- ✓ Poorly written `CGI` scripts that may allow someone to access your server will not work.

Cons:

- ✓ There are extra libraries you'll need to have in the chroot jail for `Apache` to work.
- ✓ If you use any `Perl/CGI` features with `Apache`, you will need to copy the needed binaries, `Perl` libraries and files to the appropriate spot within the chroot space. The same applies for `SSL`, `PHP`, and other third-party programs.

Necessary steps to run Apache in a chroot jail:

What we're essentially doing is creating a skeleton root file system with enough components necessary (directories, libraries, files, etc.) to allow Unix to do a chroot when the Apache daemon starts.

Step 1

The first step to do for running Apache in a chroot jail will be to set up the chroot environment, and create the root directory of the jail. We've chosen /chroot/httpd for this purpose because we want to put this on its own separate file system to prevent file system attacks. Early in our Linux installation procedure we created a special partition /chroot for this exact purpose.

```
[root@deep /]# /etc/init.d/httpd stop ← Only if Apache daemon already run.
Shutting down Apache:       [OK]

[root@deep /]# mkdir -p /chroot/httpd/dev
[root@deep /]# mkdir -p /chroot/httpd/lib
[root@deep /]# mkdir -p /chroot/httpd/etc
[root@deep /]# mkdir -p /chroot/httpd/home
[root@deep /]# mkdir -p /chroot/httpd/tmp ← Only for TLS/SSL.
[root@deep /]# chmod 777 /chroot/httpd/tmp/ ← Only for TLS/SSL.
[root@deep /]# chmod +t /chroot/httpd/tmp/ ← Only for TLS/SSL.
[root@deep /]# mkdir -p /chroot/httpd/usr/lib
[root@deep /]# mkdir -p /chroot/httpd/usr/sbin
[root@deep /]# mkdir -p /chroot/httpd/var/log
[root@deep /]# mkdir -p /chroot/httpd/var/run
```

- For Red Hat Linux 7.3 users, you should create the following additional directory:
 `[root@deep /]# mkdir /chroot/httpd/lib/i686`

We need all of the above directories because, from the point of the chroot, we're sitting at "/" and anything above this directory is inaccessible. Note that /chroot/httpd/tmp is required only if you use SSL support with Apache.

Step 2

After that, it is important to move the main configuration directory and all configuration files of Apache, the DocumentRoot directory and the httpd binary program of the web server to the chroot jail then create the special devices /dev/null and /dev/urandom which is/are require by the system to work properly. Note that /dev/urandom is requiring only if you use SSL.

```
[root@deep /]# mv /etc/httpd /chroot/httpd/etc/
[root@deep /]# mv /home/httpd /chroot/httpd/home/
[root@deep /]# mv /var/log/httpd /chroot/httpd/var/log/
[root@deep /]# mv /usr/sbin/ab /chroot/httpd/usr/sbin/
[root@deep /]# mv /usr/sbin/apxs /chroot/httpd/usr/sbin/
[root@deep /]# mv /usr/sbin/checkgid /chroot/httpd/usr/sbin/
[root@deep /]# mv /usr/sbin/dbmmanage /chroot/httpd/usr/sbin/
[root@deep /]# mv /usr/sbin/htdbm /chroot/httpd/usr/sbin/
[root@deep /]# mv /usr/sbin/htdigest /chroot/httpd/usr/sbin/
[root@deep /]# mv /usr/sbin/htpasswd /chroot/httpd/usr/sbin/
[root@deep /]# mv /usr/sbin/httpd /chroot/httpd/usr/sbin/
[root@deep /]# mv /usr/sbin/logresolve /chroot/httpd/usr/sbin/
[root@deep /]# mv /usr/sbin/rotatelogs /chroot/httpd/usr/sbin/
[root@deep /]# mknod /chroot/httpd/dev/null c 1 3
[root@deep /]# chmod 666 /chroot/httpd/dev/null
[root@deep /]# mknod /chroot/httpd/dev/urandom c 1 9 ← Only for TLS/SSL.
```

Step 3
This step is required only if you are running Apache with TLS/SSL support. In this case, you must recreate a small copy of the /usr/share/ssl directory with certs and private directories which handles the private and public keys of Apache to the chroot jail environment.

- This can be done with the following commands:

```
[root@deep /]# mkdir -p /chroot/httpd/usr/share/ssl
[root@deep /]# mkdir -p /chroot/httpd/usr/share/ssl/certs
[root@deep /]# mkdir -p /chroot/httpd/usr/share/ssl/private
[root@deep /]# chown www.www /chroot/httpd/usr/share/ssl/certs/
[root@deep /]# chown www.www /chroot/httpd/usr/share/ssl/private/
[root@deep /]# cd /usr/share/ssl/
[root@deep ssl]# mv certs/www.crt /chroot/httpd/usr/share/ssl/certs/
[root@deep ssl]# mv private/www.key /chroot/httpd/usr/share/ssl/private/
```

Step 4
Now, we must find the shared library dependencies of httpd binary and install them into the chroot structure. Use the **ldd /chroot/httpd/usr/sbin/httpd** command to find out which libraries are needed. The output (depending on what you've compiled with Apache) will be something similar to:

- To find the shared library dependencies of httpd, execute the following command:

```
[root@deep /]# ldd /chroot/httpd/usr/sbin/httpd
libaprutil.so.0 => /usr/lib/libaprutil.so.0 (0x00129000)
libapr.so.0 => /usr/lib/libapr.so.0 (0x0013b000)
libm.so.6 => /lib/libm.so.6 (0x0015a000)
libcrypt.so.1 => /lib/libcrypt.so.1 (0x0017e000)
libnsl.so.1 => /lib/libnsl.so.1 (0x001ac000)
libdl.so.2 => /lib/libdl.so.2 (0x001c0000)
libssl.so.2 => /lib/libssl.so.2 (0x001c3000)
libcrypto.so.2 => /lib/libcrypto.so.2 (0x001f1000)
libgdbm.so.2 => /usr/lib/libgdbm.so.2 (0x002c5000)
libdb-3.3.so => /lib/libdb-3.3.so (0x002cc000)
libexpat.so.0 => /usr/lib/libexpat.so.0 (0x00352000)
libpthread.so.0 => /lib/libpthread.so.0 (0x00372000)
libc.so.6 => /lib/libc.so.6 (0x003a2000)
libgcc_s.so.1 => /lib/libgcc_s.so.1 (0x004f6000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x00110000)
```

What we can see here is the fact that depending of what programs have been compiled and included with Apache, the shared library dependencies may differ.

Step 5
Once the required libraries have been identified, copy them to the appropriate location into the chroot jail. In our example these are the shared libraries identified above.

```
[root@deep /]# cp /usr/lib/libaprutil.so.0 /chroot/httpd/usr/lib/
[root@deep /]# cp /usr/lib/libapr.so.0 /chroot/httpd/usr/lib/
[root@deep /]# cp /usr/lib/libgdbm.so.2 /chroot/httpd/usr/lib/
[root@deep /]# cp /usr/lib/libexpat.so.0 /chroot/httpd/usr/lib/
[root@deep /]# cp /lib/libm.so.6 /chroot/httpd/lib/
[root@deep /]# cp /lib/libcrypt.so.1 /chroot/httpd/lib/
[root@deep /]# cp /lib/libnsl.so.1 /chroot/httpd/lib/
[root@deep /]# cp /lib/libdl.so.2 /chroot/httpd/lib/
[root@deep /]# cp /lib/libssl.so.2 /chroot/httpd/lib/
[root@deep /]# cp /lib/libcrypto.so.2 /chroot/httpd/lib/
[root@deep /]# cp /lib/libdb-3.3.so /chroot/httpd/lib/
[root@deep /]# cp /lib/libpthread.so.0 /chroot/httpd/lib/
[root@deep /]# cp /lib/libc.so.6 /chroot/httpd/lib/
[root@deep /]# cp /lib/libgcc_s.so.1 /chroot/httpd/lib/
[root@deep /]# cp /lib/ld-linux.so.2 /chroot/httpd/lib/
```

You'll also need the following extra libraries for some network functions, like resolving:

```
[root@deep /]# cp /lib/libnss_compat* /chroot/httpd/lib/
[root@deep /]# cp /lib/libnss_dns* /chroot/httpd/lib/
[root@deep /]# cp /lib/libnss_files* /chroot/httpd/lib/
[root@deep /]# strip -R .comment /chroot/httpd/lib/*
[root@deep /]# strip -R .comment /chroot/httpd/usr/lib/*
```

- For Red Hat Linux 7.3 users, you should copy the following additional library:
 `[root@deep /]# cp /lib/i686/libc.so.6 /chroot/httpd/lib/i686/`

NOTE: The "`strip -R .comment`" commands will remove all the named section "`.comment`" from the libraries files under the `/usr/lib` and `/lib` directory of the chroot jail and will make them smaller in size to help in performance of them.

Step 6
Now we need to copy the **`passwd`** and **`group`** files inside the `/chroot/httpd/etc` directory. Next, we'll remove all entries except for the user that `httpd` runs as in both files.

```
[root@deep /]# cp /etc/passwd /chroot/httpd/etc/
[root@deep /]# cp /etc/group /chroot/httpd/etc/
```

- Edit the **`passwd`** file under the chroot jail (`vi /chroot/httpd/etc/passwd`) and delete all entries except for the user `httpd` run as (in our configuration, it's "`www`"):

```
www:x:48:48:Web Server:/home/httpd:/bin/false
```

- Edit the **`group`** file under the chroot jail (`vi /chroot/httpd/etc/group`) and delete all entries except the group `httpd` run as (in our configuration it's "`www`"):

```
www:x:48:
```

Step 7
You will also need **resolv.conf**, **nsswitch.conf**, **localtime** and **hosts** files in your chroot jail structure.

```
[root@deep /]# cp /etc/resolv.conf /chroot/httpd/etc/
[root@deep /]# cp /etc/nsswitch.conf /chroot/httpd/etc/
[root@deep /]# cp /etc/localtime /chroot/httpd/etc/
[root@deep /]# cp /etc/hosts /chroot/httpd/etc/
```

Step 8
Now we must set some files in the chroot jail directory immutable for better security.

- These procedures can be accomplished with the following commands:

```
[root@deep /]# cd /chroot/httpd/etc/
[root@deep etc]# chattr +i passwd
[root@deep etc]# chattr +i group
[root@deep etc]# chattr +i resolv.conf
[root@deep etc]# chattr +i hosts
[root@deep etc]# chattr +i nsswitch.conf
```

WARNING: Don't forget to remove the immutable bit on these files if you have some modifications to bring to them with the command "`chattr -i`".

Step 9
With all modifications for running `Apache` in chroot jail, the `Apache` logs files resides now in the `/chroot/httpd/var/log/httpd` directory instead of `/var/log/httpd`, and for this reason we need to modify the existing `/etc/logrotate.d/httpd` file to point to the new chrooted directory.

- Edit the **httpd** file (`vi /etc/logrotate.d/httpd`) and add/change the lines:

```
/chroot/httpd/var/log/httpd/access_log {
    missingok
    postrotate
        /usr/bin/killall -HUP /chroot/httpd/usr/sbin/httpd
    endscript
}

/chroot/httpd/var/log/httpd/error_log {
    missingok
    postrotate
        /usr/bin/killall -HUP /chroot/httpd/usr/sbin/httpd
    endscript
}

/chroot/httpd/var/log/httpd/ssl_request_log {
    missingok
    postrotate
        /usr/bin/killall -HUP /chroot/httpd/usr/sbin/httpd
    endscript
}
```

Step 10
The default `httpd` initialization script file of `Apache` starts the daemon "`httpd`" outside the chroot jail. We must change it now to start `httpd` from the chroot jail environment.

Since there are many lines to modify from the original initialization script file of `Apache` to make it start in the jail environment, I decided to make a new initialization file as shown below. Each line in bold are the one that are different from the original script file. In this way you'll be able to see how I made it.

- Edit the **`httpd`** script file (`vi /etc/init.d/httpd`) and add/change the lines:

```
#!/bin/bash

# This shell script takes care of starting and stopping Apache.
#
# chkconfig: 345 85 15
# description: Apache is a World Wide Web server. It is used to serve \
#              HTML files and CGI.
#
# processname: httpd
# config: /chroot/httpd/etc/httpd/httpd.conf
# pidfile: /chroot/httpd/var/run/httpd.pid

# Source function library.
. /etc/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Source for additional options if we have them.
if [ -f /etc/sysconfig/httpd ] ; then
        . /etc/sysconfig/httpd
fi

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

# If Apache is not available stop now.
[ -f /chroot/httpd/usr/sbin/httpd ] || exit 0

RETVAL=0
prog="Apache"

start() {
        echo -n $"Starting $prog: "
        /usr/sbin/chroot /chroot/httpd/ /usr/sbin/httpd $OPTIONS
        RETVAL=$?
        echo
        [ $RETVAL = 0 ] && touch /var/lock/subsys/httpd
        return $RETVAL
}

stop() {
        echo -n $"Shutting down $prog: "
        kill -TERM `cat /chroot/httpd/var/run/httpd.pid`
        RETVAL=$?
        echo "  [ OK ]"
        [ $RETVAL = 0 ] && rm -f /var/lock/subsys/httpd
        return $RETVAL
}
```

```
# See how we were called.
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  status)
        status /chroot/httpd/usr/sbin/httpd
        RETVAL=$?
        ;;
  restart)
        kill -USR1 `cat /chroot/httpd/var/run/httpd.pid`
        RETVAL=$?
        ;;
  condrestart)
        if [ -f /var/lock/subsys/httpd ]; then
        kill -USR1 `cat /chroot/httpd/var/run/httpd.pid`
        RETVAL=$?
        fi
        ;;
  *)
        echo $"Usage: $0 {start|stop|status|restart|condrestart}"
        exit 1
esac
exit $RETVAL
```

Step 11
Finally, we must test the new chrooted jail configuration of our web server.

- Start the new chrooted jail `Apache` with the following command:
  ```
  [root@deep /]# /etc/init.d/httpd start
  Starting Apache:           [OK]
  ```

- If you don't get any errors, do a **`ps ax | grep httpd`** and see if we're running:
  ```
  [root@deep /]# ps ax | grep httpd
  8098 ?        S      0:00 /usr/sbin/httpd -DSSL
  ```

If so, lets check to make sure it's chrooted by picking out its process number and doing **`ls -la /proc/that_process_number/root/`**.

```
[root@deep /]# ls -la /proc/8098/root/
```

If you see something like:

```
drwxr-xr-x    9 root     root         4096 Jun  5 23:03 ./
drwxr-xr-x    6 root     root         4096 Jun  5 23:02 ../
drwxr-xr-x    2 root     root         4096 Jun  5 23:04 dev/
drwxr-xr-x    3 root     root         4096 Jun  5 23:22 etc/
drwxr-xr-x    3 root     root         4096 Jun  5 23:04 home/
drwxr-xr-x    2 root     root         4096 Jun  5 23:16 lib/
drwxrwxrwt    2 root     root         4096 Jun  5 23:02 tmp/
drwxr-xr-x    5 root     root         4096 Jun  5 23:03 usr/
drwxr-xr-x    4 root     root         4096 Jun  5 23:03 var/
```

Congratulations!

Running Apache with users authentication support

These steps are necessary only if you think that you'll use an access file authentication system for your web site. Access file authentication is used when you are in the need to protect some part of your web site with a user password. With Apache, a lot of options and modules exist to protect your site with usernames and passwords.

Of course other methods to implement authentication via a web server exist with programming language like PHP, CGI, C, etc but here we talk about the way to do it with what comes with Apache via modules support. Three build in modules with Apache allow us to archive this result; there are "auth_module", "auth_dbm_module", and "auth_digest_module". Each one have some advandatage and disadvantage compared to other and in our example we choose to explain and show you how to enable user authentication support with the Apache module called "auth_dbm_module" because it gives us a good average between security, performance and flexibility.

If you prefer to go with another Apache module for user authentication, I recommend you to read the Apache manual for its implementation on your web server. The concept is the same as for the one we explain you here and only configuration lines change.

Step 1
The dbmmanage program utility, which comes by default with Apache, can be used to create and update usernames and passwords of HTTP users. This method use a DBM format files that is the fastest mechanism when you have thousands users to manage in your password file. First of all, it's important to change the permission of this program to be (0510/-r-x--x---), writable only by the super-user "root", readable and executable by group and nothing for the others.

- To change the permissions on the dbmmanage program, use the following command:
```
[root@deep /]# chmod 510 /usr/sbin/dbmmanage
```

Step 2
Once the permission has been set to this program, we can create the DBM format file with username and password.

- To create a username and password, use the following command:
```
[root@deep /]# /usr/sbin/dbmmanage  /etc/httpd/dbmpasswd adduser gmourani
New password:
Re-type new password:
User gmourani added with password encrypted to dtkTL83yvMbFQ using crypt
```

Where </etc/httpd/> is the location where we want to create and handle this password file, <dbmpasswd> is the name we give to the password file, and <gmourani> is the username of the user we want to add in our dbmpasswd file.

WARNING: Every user that we would like to add to the dbmpasswd file doesn't need to be a real user on the system. I mean that it is not necessary to have them in the /etc/passwd file.

Step 3
If you use the dbmmanage utility of Apache web server to create passwords and usernames, don't forget to include in your /etc/httpd/httpd.conf configuration file the part of your web site you need to protect with user password authentication.

- Edit the **httpd.conf** file (vi /etc/httpd/httpd.conf) and add the following lines to protect the "private" directory of your web site with user password authentication:

```
<Directory "/home/httpd/html/private">
    Options None
    AllowOverride AuthConfig
    AuthName "Restricted Section"
    AuthType Basic
    AuthDBMUserFile /etc/httpd/dbmpasswd
    require valid-user
</Directory>
```

The path </home/httpd/html/private> specifies the web directory we want to protect with a password and username, the </etc/httpd/dbmpasswd> specifies the location of the DBM password file on the server.

Step 4
As for any other modules with Apache, we have to activate the "mod_auth_dbm.so" module for the web server to support it. This is possible by uncommenting the line related to the module in question insdide the httpd.conf file.

- Edit the **httpd.conf** file (vi /etc/httpd/httpd.conf), and change the line:

```
#LoadModule auth_dbm_module     lib/apache/mod_auth_dbm.so
```

To read:

```
LoadModule auth_dbm_module     lib/apache/mod_auth_dbm.so
```

Step 5
Once the above lines have been included/uncommented into the httpd.conf file of Apache to enable user's password authentication feature, you must restart Apache for the changes to take effect.

- To restart Apache, use the following command:

```
[root@deep /]# /etc/init.d/httpd restart
Shutting down Apache:         [OK]
Starting Apache:              [OK]
```

Step 6
Finally, we must test the new protected web directory called "private". To verify that it works, points your browser to the following address: http://www.domain.com/private/. The <www.domain.com> is the address where your Apache web server lives and </private/> is the directory protected with user password authentication.

Caching frequently requested static files

There is a special module with the Apache distribution called "mod_file_cache" that can by used to improve the performance of your web server. This module works by providing mappings of a statically configured list of frequently requested, but not changed, files in your RootDirectory. Therefore, if files displayed by Apache don't change often, you can use this useful module to memory-map the static documents and increase the speed of your web server. This means visitors to your sites get faster download times. This module should be used with care because you can easily create a broken site.

Step 1
The magical command to map all files under a RootDirectory to a specific text file of your choice is shown below. Once again, this Apache module is only useful when you have a static web site, I mean by static, a web site where contents do not change often.

- To memory-map static documents, use the following command:
  ```
  [root@deep /]# find /home/httpd/html -type f -print | sed -e
  's/.*/mmapfile &/' > /etc/httpd/mmap.conf
  ```

The </home/httpd/html> is the RootDirectory, or to be more precise, the directory out of which you will serve your documents, and the </etc/httpd/mmap.conf> is the location where we want to create this file "mmap.conf" that contains a static memory-map of all documents under our RootDirectory.

WARNING: If you add or update contents into your site, don't forget to reuse this command line again and restart you web server for the changes to take effect. A cron job to automate the task is a good idea.

Step 2
Once the "mmap.conf" file has been create under the location where we have chosen to keep this file, we must include it in the httpd.conf file for Apache to be able to use its interesting features on our server.

- Edit the **httpd.conf** file (vi /etc/httpd/httpd.conf) and add/check the lines:

```
<IfModule mod_file_cache.c>
<IfModule mod_include.c>
    Include /etc/httpd/mmap.conf
</IfModule>
</IfModule>
```

NOTE: See your Apache documentation for more information about the use of mod_file_cache. Remember that this feature must be used only when you serve documents that don't change often on your web site.

Step 3
As for any other modules with Apache, we have to activate the "mod_file_cache.so" module for the web server to support it. This is possible by uncommenting the line related to the module in question insdide the httpd.conf file.

- Edit the **httpd.conf** file (vi /etc/httpd/httpd.conf), and change the line:

  ```
  #LoadModule file_cache_module    lib/apache/mod_file_cache.so
  ```

 To read:

  ```
  LoadModule file_cache_module    lib/apache/mod_file_cache.so
  ```

Step 4
Finally, the last step to do is to restart the Apache web server for the changes to take effect:

- To restart Apache, use the following command:

  ```
  [root@deep /]# /etc/init.d/httpd restart
  Shutting down Apache:        [OK]
  Starting Apache:             [OK]
  ```

Some statistics about Apache and Linux

People like to see statistics and benchmark of different kind. It is always interesting to know the last milliseconds, bits we can take from our software and servers. The following pages explain and show you another one about Apache and Linux but not in the way you are accustomed in general. The moral is that: it is not always good to try or trust benchmarks, technologies limit, unthinking factor, etc that may influence results, but stability of your system is something you must have and keep.

What are some of the actual facts that the tests came up with?
With 1 CPU and 256 MB RAM, Linux & Apache achieved 1,314 http requests per second.

First of, let's just look at an approximation of the situation that this represents:
1,314 hits/sec * 3600 sec/hour * 24 hours/day = 113,529,600 hits/day.

So Linux/Apache should be able to handle your site on a 1 CPU 256 MB RAM machine if you get 113 million hits per day or less. Of course, this only works if your access is 100% even, which is extremely unrealistic. Let's assume that your busy times get ten times more hits per second than your average hits/second. That means that a single CPU Linux machine with 256 meg of RAM should work for you if you get about 11 million hits every day (113/10 = 11.3).

Heck, let's be more conservative. Let's say that your busy times get 100 times more hits/second than your average hits/second. That means that if you get 1.1 million hits per day or less, that same machine will serve your site just fine (113/100 = 1.13).

OK, there's that way of looking at it, but it's not really a good way. It's a very coarse approximation of access patterns and what a site needs. Let's try another way of looking at this. Let's do some simple calculations to see what sort of bandwidth these numbers mean. Bandwidth will be a better and more constant method of determining whom these numbers apply to than guessed at hit ratios.

The files served must be of "varying sizes", so we'll have to make some assumptions about the average size of the files being served. Since over 1000 files were served per second, it is pretty safe to work by averages.

Some numbers:

- 1,314 hits/sec * 1 kilobyte/hit * 8192 bits/kilobyte = 10764288 bits/sec = 10 MBits/sec.
- 1,314 hits/sec * 2 kilobytes/hit * 8192 bits/kilobyte = 21528576 bits/sec = 21 MBits/sec.
- 1,314 hits/sec * 5 kilobytes/hit * 8192 bits/kilobyte = 53821440 bits/sec = 53 MBits/sec.
- 1,314 hits/sec * 10 kilobytes/hit * 8192 bits/kilobyte = 107642880 bits/sec = 107 MBits/sec.
- 1,314 hits/sec * 25 kilobytes/hit * 8192 bits/kilobyte = 269107200 bits/sec = 269 MBits/sec.

Just as a reference, a T1 line is worth approximately 1.5 MBits/sec, these numbers don't include `TCP/IP` & `HTTP` overhead.

Now, what does this tell us? Well, that if you are serving up 1,314 pages per second where the average page is only 1 kilobyte, you'll need ten (10) T1 lines or the equivalent until the computer is the limiting factor. What site on earth is going to be getting a sustained >1000 hits per second for 1 kilobyte files? Certainly not one with any graphics in it.

Let's assume that you're running a site with graphics in it and that you're average file is 5 kilobytes - not too conservative or too liberal. This means that if you're serving up 1,314 of them a second, you'll need 53 MBits of bandwidth. And there are no peak issues here; you can't peak out more than your bandwidth.

Let's go at it another way, this time starting with our available bandwidth:

1 T1 Line * 1.5 MBits/T1 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/kilobyte = 184 hits/sec.
1 T1 Line * 1.5 MBits/T1 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/2 kilobytes = 92 hits/sec.
1 T1 Line * 1.5 MBits/T1 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/5 kilobytes = 37 hits/sec.
1 T1 Line * 1.5 MBits/T1 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/10 kilobytes = 19 hits/sec.
1 T1 Line * 1.5 MBits/T1 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/25 kilobytes = 8 hits/sec.

5 T1 Line * 1.5 MBits/T1 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/kilobyte = 916 hits/sec.
5 T1 Line * 1.5 MBits/T1 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/2 kilobytes = 458 hits/sec.
5 T1 Line * 1.5 MBits/T1 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/5 kilobytes = 183 hits/sec.
5 T1 Line * 1.5 MBits/T1 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/10 kilobytes = 92 hits/sec.
5 T1 Line * 1.5 MBits/T1 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/25 kilobytes = 36 hits/sec.

1 T3 Line * 45 MBits/T3 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/kilobyte = 5,494 hits/sec.
1 T3 Line * 45 MBits/T3 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/2 kilobytes = 2747 hits/sec.
1 T3 Line * 45 MBits/T3 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/5 kilobytes = 1099 hits/sec.
1 T3 Line * 45 MBits/T3 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/10 kilobytes = 550 hits/sec.
1 T3 Line * 45 MBits/T3 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/25 kilobytes = 220 hits/sec.

1 OC3 Line * 155 MBits/OC3 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/kilobyte = 18,921 hits/sec.
1 OC3 Line * 155 MBits/OC3 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/2 kilobytes = 9461 hits/sec.
1 OC3 Line * 155 MBits/OC3 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/5 kilobytes = 3785 hits/sec.
1 OC3 Line * 155 MBits/OC3 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/10 kilobytes = 1,893 hits/sec.
1 OC3 Line * 155 MBits/OC3 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/25 kilobytes = 757 hits/sec.

NOTE: These numbers don't include `TCP/IP` or `HTTP` overhead.

It is clear that the numbers are only significant when you have the equivalent bandwidth of over 6 T1 lines. Let's be clear about this: if you have only **five (5) T1 lines** or less, a single CPU Linux machine with 256 MB RAM will **wait on your internet connection** and not be able to serve up to its full potential.

Let me re-emphasize this: A single CPU Linux machine with 256 MB RAM running Apache will **run faster than your internet connection**! Put another way, if your site runs on five (5) T1 lines or less, a single CPU Linux machine with 256 MB RAM will **more than fulfill your needs with CPU cycles left over**.

Let's make an assumption that you either (a) have pages with more than about a screen of text or (b) black and white pictures that make your average file size 5K. Given this, would indicate that a single CPU Linux machine with only 256 MB RAM running Apache would be **constantly waiting on your T3 line**. In other words, a single CPU Linux machine with 256 MB RAM will **serve your needs with room to grow** if your site is served by a **T3 line** or less.

One might also conclude that if you serve things like color pictures (other than small buttons and doodads) and thus your average file size is 25K, a single CPU Linux machine with 256 MB RAM will serve your site just fine even if you are served by an OC3 line that you have all to your self.

Further documentation

For more details, there are some manual pages about Apache that you could read:

```
$ man dbmmanage (1)          - Create and update user authentication files in DBM format.
$ man htdigest (1)           - Create and update user authentication files.
$ man htpasswd (1)           - Create and update user authentication files.
$ man ab (8)                 - Apache HTTP server benchmarking tool.
$ man httpd (8)              - Apache hypertext transfer protocol server.
$ man logresolve (8)         - Resolve hostnames for IP-addresses in Apache logfiles.
$ man rotatelogs (8)         - Rotate Apache logs without having to kill the server.
```

CHAPTER 44

PHP

IN THIS CHAPTER

1. **Compiling - Optimizing & Installing `PHP`**
2. **Configuring `PHP`**
3. **Running `PHP` in a chroot jail**
4. **Running `PHP` with the `PHP` Accelerator program**

Linux PHP

Abstract

This chapter is a chapter related to the Apache web server, you should read it only if you have installed Apache on your system and want to provide and make it run with some additional feature. Here we talk about PHP with Apache.

Everyone using a web server knows about PHP and its possibilities. It seems that PHP will certainly replace other language like Perl or CGI for web services in the future. This is due to its simplicity of use and many developers known about this and has already developed software that run with PHP on a web server. When you need to add some popular web service to your web server, you will inevitably find that PHP is required and that you need to install it with Apache. In this chapter we discuss about the way to integrate it with Apache as a module because we already have installed Apache in the previous chapter with modules support.

In regard to the previous book, I've decided to explain you how to compile and install PHP with most interesting third party services like MySQL, PostgreSQL, LDAP, IMAP, and SSL support. This will let us enable or disable which service and feature we want to provide with PHP for our web server in an easy way without the need to recompile the software every time we decide to add or remove features. From the point of view of performance, there is no so big difference if we run PHP as a module with Apache.

Building PHP as a module support into the Apache web server has some interesting advantage because we can easily upgrade the software when required without the need to rebuild the whole Apache web server.

PHP (recursive acronym for "PHP: **H**ypertext **P**reprocessor") is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML. PHP is mainly focused on server-side scripting, so you can do anything any other CGI program can do, such as collect form data, generate dynamic page content, or send and receive cookies. But PHP can do much more. Just look on the Internet for the myriad of open source software available in PHP language.

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, at personal discretion).
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest PHP version number is 4.2.1

The procedures given in this chapter are likely to work on all Linux platforms, but we have only tested it on OpenNA Linux and Red Hat Linux.

Packages

The following is based on information listed by PHP as of 2002/05/13. Please regularly check http://www.php.net/ for the latest status. We chose to install the required component from a source file because it provides the facility to fine tune the installation.

Source code is available from:
PHP Homepage: http://www.php.net/
You must be sure to download: php-4.2.1.tar.gz

Prerequisites

PHP requires that the software below is already installed on your system to be able to compile successfully. If this is not the case, you must install it. Please make sure you have this program installed on your machine before you proceed with this chapter.

- ✓ Apache is required to be able to use PHP in your system.
- ✓ OpenSSL is required to be able to use PHP with SSL support in your system.
- ✓ imap-devel is required to be able to build PHP in your system.
- ✓ mysql is required to be able to use PHP in your system.
- ✓ mysql-devel is required to be able to build PHP in your system.
- ✓ postgresql package is required to be able to use PHP in your system.
- ✓ postgresql-devel package is required to be able to build PHP in your system.
- ✓ openldap package is required to be able to use PHP in your system.
- ✓ openldap-devel package is required to be able to build PHP in your system.
- ✓ bzip2-devel package is required to be able to build PHP in your system.
- ✓ libpng package is required to be able to use PHP in your system.
- ✓ libpng-devel package is required to be able to build PHP in your system.
- ✓ libjpeg package is required to be able to use PHP in your system.
- ✓ libjpeg-devel package is required to be able to build PHP in your system.
- ✓ freetype package is required to be able to use PHP in your system.
- ✓ freetype-devel package is required to be able to build PHP in your system.
- ✓ gd package is required to be able to use PHP in your system.
- ✓ gd-devel package is required to be able to build PHP in your system.
- ✓ aspell package is required to be able to use PHP in your system.

- ✓ pspell package is required to be able to use PHP in your system.
- ✓ pspell-devel package is required to be able to build PHP in your system.
- ✓ perl package is required to be able to use PHP in your system.
- ✓ file package is required to be able to use PHP in your system.
- ✓ dmalloc package is required to be able to use PHP in your system.
- ✓ gmp package is required to be able to use PHP in your system.
- ✓ gmp-devel package is required to be able to use PHP in your system.
- ✓ zlib-devel package is required to be able to use PHP in your system.
- ✓ pam-devel package is required to be able to use PHP in your system.

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all the files installed on the system in the eventuality of an update in the future. To solve the problem, it is a good idea to make a list of files on the system before you install PHP, and then one afterwards, and then compare them using the **diff** utility to find out what files were placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep root]# find /* > PHP1
  ```
- And the following one after you install the software:
  ```
  [root@deep root]# find /* > PHP2
  ```
- Then use the following command to get a list of what changed:
  ```
  [root@deep root]# diff PHP1 PHP2 > PHP-Installed
  ```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. In the example above, we use the /root directory of the system to store all generated list files.

Compiling - Optimizing & Installing PHP

Below are the steps that you must make to configure, compile and optimize the PHP software before installing it onto your system. First off, we install the program as the user "root" so as to avoid permissioning problems.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- This can be done with the following commands:
  ```
  [root@deep /]# cp php-version.tar.gz /var/tmp/
  [root@deep /]# cd /var/tmp/
  [root@deep tmp]# tar xzpf php-version.tar.gz
  ```

Step 2
After that, move into the newly created PHP source directory and perform the following steps to configure and optimize PHP for your system.

- To move into the newly created PHP source directory use the command:
 [root@deep tmp]# **cd php-4.2.1/**

Step 3
There is one PHP source file to modify before going in configuration and compilation of the program; the change allows us to fix a problem related to PostgreSQL library file. PHP suppose by default that PostgreSQL library is available as a dynamically loadable library only and do not take in consideration that we may provide the PostgreSQL libraries as static library.

To solve the problem, we have to edit PHP source file related to PostgreSQL and change one line inside the code to inform it that our PostgreSQL library is available as a static library for the software to be able to find and use it during compilation of the program. I know that the hack is not really clean but it work fine and this is what we want.

- Edit the **config.m4** file (vi +30 ext/pgsql/config.m4) and change the line:

  ```
  if test -f "$i/$j/libpq.so"; then
  ```

 To read:

  ```
  if test -f "$i/$j/libpq.a"; then
  ```

Step 4
Once the required modification has been made into the related source file of PHP, it is time configure and optimize it for our system. As you will see further down, in our compilation of PHP, we disable any unneeded modules and enable support for IMAP, IMAP with SSL, MySQL, PostgreSQL, and LDAP together.

This is a good practice even if you don't use all of these features with PHP because everything are compiled as a modules and will become active only if you enable the related module in question inside your php.ini file.

- To compile and optimize PHP use the following compilation lines:

```
CFLAGS="-O2 -march=i686 -funroll-loops -fPIC"; export CFLAGS
LIBS="-lttf -lfreetype -lpng -ljpeg -lz -lnsl"; export LIBS
EXTENSION_DIR=/usr/lib/php4; export EXTENSION_DIR
IMAP_SHARED_LIBADD=-lc-client ; export IMAP_SHARED_LIBADD

./buildconf
./configure \
--prefix=/usr \
--with-layout=GNU \
--with-apxs2 \
--with-config-file-path=/etc/httpd \
--with-exec-dir=/usr/bin \
--with-openssl \
--with-zlib \
--with-bz2 \
--with-gd \
--with-ttf \
```

```
--with-png \
--with-jpeg-dir=/usr \
--with-png-dir=/usr \
--with-freetype-dir=/usr \
--with-expat-dir=/usr \
--with-gmp \
--with-xml \
--with-imap=shared \
--with-imap-ssl \
--with-mysql=shared \
--with-mysql-sock=/var/lib/mysql/mysql.sock \
--with-pgsql=shared \
--with-ldap=shared \
--with-pspell \
--disable-debug \
--disable-posix \
--disable-rpath \
--enable-safe-mode \
--enable-magic-quotes \
--enable-dmalloc \
--enable-bcmath \
--enable-dio \
--enable-gd-native-ttf \
--enable-sysvsem \
--enable-sysvshm \
--enable-wddx \
--enable-versioning \
--enable-pic \
--enable-inline-optimization \
--enable-memory-limit
```

Step 5
At this stage the program is ready to be built and installed. We build `PHP` with the '`make`' command and produce a list of files on the system before we install the software, and one afterwards, then compare them using the **`diff`** utility to find out what files were placed where and finally install `PHP`.

```
[root@deep php-4.2.1]# make
[root@deep php-4.2.1]# cd
[root@deep root]# find /* > PHP1
[root@deep root]# cd /var/tmp/php-4.2.1/
[root@deep php-4.2.1]# make install
[root@deep php-4.2.1]# install -m0644 php.ini-dist /etc/httpd/php.ini
[root@deep php-4.2.1]# strip -R .comment /usr/lib/php4/*.so
[root@deep php-4.2.1]# cd
[root@deep root]# find /* > PHP2
[root@deep root]# diff PHP1 PHP2 > PHP-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

Step 6
Once the compilation, optimization and installation of the software has completed, we can free up some disk space by deleting the program tar archive and the related source directory, since they are no longer needed.

- To delete Apache and its related source directory, use the following commands:
```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf php-version/
[root@deep tmp]# rm -f php-version.tar.gz
```

The rm command as used above will remove all the source files we have used to compile and install PHP. It will also remove the PHP compressed archive from the /var/tmp directory.

Configuring PHP

After PHP has been built and installed successfully on your system, the next step is to configure and customize its configuration file to fit your needs.

- ✓ /etc/httpd/php.ini: (The PHP Configuration File)

/etc/httpd/php.ini: The PHP Configuration File

The php.ini file is the main configuration file for the PHP hypertext preprocessor. A lot options exist, and it's important to read the documentation that comes with PHP for more information on different settings and parameters.

Step 1
The following configuration is a full secure working configuration file for PHP. Also, it's important to note that I only comment parameters that relate to security and optimization, and leave all the others to your own research. Text in bold is the parts of the configuration file that must be customized and adjusted to satisfy your needs.

- Edit the **php.ini** file (vi /etc/httpd/php.ini) and set your needs:

```
[PHP]

; Language Options
engine                          = On
short_open_tag                  = On
asp_tags                        = Off
precision                       = 14
y2k_compliance                  = Off
output_buffering                = Off
output_handler                  =
unserialize_callback_func       =
zlib.output_compression         = On
implicit_flush                  = Off
allow_call_time_pass_reference  = Off


; Safe Mode
safe_mode                       = On
safe_mode_gid                   = Off
safe_mode_include_dir           = /var/lib/mysql
safe_mode_exec_dir              =
;open_basedir                   =
safe_mode_allowed_env_vars      = PHP_
safe_mode_protected_env_vars    = LD_LIBRARY_PATH
```

```
disable_functions               =

; Font Colors
highlight.string                = #CC0000
highlight.comment               = #FF9900
highlight.keyword               = #006600
highlight.bg                    = #FFFFFF
highlight.default               = #0000CC
highlight.html                  = #000000

; Misc
expose_php                      = Off

; Resource Limits
max_execution_time              = 30
memory_limit                    = 8M

; Error handling and logging
error_reporting                 = E_ALL
display_errors                  = Off
display_startup_errors          = Off
log_errors                      = On
track_errors                    = Off
html_errors                     = Off
error_log                       = syslog
warn_plus_overloading           = Off

; Data Handling
;arg_separator.output           = "&amp;" ; Default is "&".
;arg_separator.input            = ";&" ; Default is "&".
variables_order                 = "GPCS"
register_globals                = Off
register_argc_argv              = On
post_max_size                   = 8M

; Magic Quotes
magic_quotes_gpc                = Off
magic_quotes_runtime            = Off
magic_quotes_sybase             = Off
auto_prepend_file               =
auto_append_file                =
default_mimetype                = "text/html"
default_charset                 = "iso-8859-1"
;always_populate_raw_post_data  = On

; Paths and Directories
;include_path                   = ".:/php/includes"
doc_root                        =
user_dir                        =
extension_dir                   = /usr/lib/php4
enable_dl                       = Off
; cgi.force_redirect            = On
; cgi.redirect_status_env       =

; File Uploads
```

```
file_uploads                    = Off
;upload_tmp_dir                =
upload_max_filesize             = 1M


; Fopen wrappers
allow_url_fopen                 = On
;from                           = "anonymous@domain.com"


; Dynamic Extensions
;extension                      = imap.so
;extension                      = ldap.so
;extension                      = mysql.so
;extension                      = pgsql.so


; PHP Accelerator extension
;zend_extension="/usr/lib/php4/php_accelerator_1.3.1pre3.so"
;phpa.registration_key = 28ccb4e0144fa5f409fbfb3834b47358


[Syslog]
define_syslog_variables         = Off
;sendmail_path                  =


[SQL]
sql.safe_mode                   = Off


[ODBC]
odbc.allow_persistent           = Off
odbc.check_persistent           = On
odbc.max_persistent             = -1
odbc.max_links                  = -1
odbc.defaultlrl                 = 4096
odbc.defaultbinmode             = 1


[MySQL]
mysql.allow_persistent          = Off
mysql.max_persistent            = -1
mysql.max_links                 = -1
mysql.default_port              =
mysql.default_socket            = /var/lib/mysql/mysql.sock
mysql.default_host              =
mysql.default_user              =
mysql.default_password          =


[PostgresSQL]
pgsql.allow_persistent          = Off
pgsql.auto_reset_persistent     = Off
pgsql.max_persistent            = -1
pgsql.max_links                 = -1


[bcmath]
bcmath.scale                    = 0


[browscap]
```

```
;browscap                       = extra/browscap.ini


[Session]
session.save_handler            = files
session.save_path               = /tmp
session.use_cookies             = 1
session.name                    = PHPSESSID
session.auto_start              = 0
session.cookie_lifetime         = 0
session.cookie_path             = /
session.cookie_domain           =
session.serialize_handler       = php
session.gc_probability          = 1
session.gc_maxlifetime          = 1440
session.referer_check           =
session.entropy_length          = 0
session.entropy_file            =
;session.entropy_length         = 16
;session.entropy_file           = /dev/urandom
session.cache_limiter           = nocache
session.cache_expire            = 180
session.use_trans_sid           = 0
url_rewriter.tags               =
"a=href,area=href,frame=src,input=src,form=fakeentry"


[Assertion]
;assert.active                  = On
;assert.warning                 = On
;assert.bail                    = Off
;assert.callback                = 0
;assert.quiet_eval              = 0


[Sockets]
sockets.use_system_read         = On
```

This tells the `php.ini` file to set itself up for this particular configuration setup with:

`engine = On`
This directive "`engine`" is used by sites that would like to turn `PHP` parsing on and off on a per-directory or per-virtual server basis. By putting engine off in the appropriate places in the `httpd.conf` file, `PHP` can be enabled or disabled with the `Apache` module version of `PHP`. In most cases, we should keep the default setting of "`On`" here or you really don't need to use `PHP`.

`short_open_tag = On`
This directive "`short_open_tag`" is used to inform the `PHP` software whether the short form (`<? ?>`) of `PHP`'s open tag should be allowed or not on the server. It is important to note that if you want to use `PHP` in combination with `XML` feature, you have to disable this option. If disabled, you must use the long form of the open tag (`<?php ?>`). In most case, we can keep the default value of "`On`" here. Change to "`Off`" only if you know what you do and what `PHP` external software you use with your web server.

`asp_tags = Off`
This directive "`asp_tags`" is used to enables the use of `ASP`-like `<% %>` tags in addition to the usual `<?php ?>` tags. You should say "`On`" here only if you use some `ASP` like language with `PHP` on your Unix system. Remember that `ASP` comes from Microsoft; therefore use this feature with `PHP` only if you want to run some `ASP` script files that come from Microsoft on Unix.

precision = 14
This directive “precision” is used to define the number of significant digits displayed in floating point numbers with PHP. The default value of "12" is ok for most of us and should be changed only if you have some good reason to do it.

y2k_compliance = Off
This directive “y2k_compliance” is used to enforce year 2000 compliance with PHP on your web server. It’s important to note that changing this feature to "On" will cause problems with non-compliant browsers. Therefore I recommend you to keep the default setting of "Off" here.

output_buffering = Off
This directive “output_buffering” is used to enable output buffering for all files with PHP. This allows PHP to send header lines (including cookies) even after sending body content, at the price of slowing PHP's output layer a bit. For performance reason, I recommend you to keep the default setting of "Off" here. In general, this doesn't pose any problem with external software using this function because you or the author of the software can enable output buffering during runtime by calling the output. This is a performance feature.

output_handler =
This directive “output_handler” is used to redirect all of the output of your scripts to a function. This can be used for example to transparently compress PHP output before sending it to a browsers that support gzip or deflate encoding. It's important to note that setting this option to an output handler will automatically turns “On” the above "output_buffering" directive. For compatibility reason with available browsers on the Net and to save CPU loads and resources on your server, I recommend you to keep this directive with its default empty value. This is a performance feature.

unserialize_callback_func =
This directive “unserialize_callback_func” is used to call a unserialize callback function defined as the value of this directive. In general, we can keep the default empty value here. Only developers and advanced users who know when they should define and use this directive should change the default setting.

zlib.output_compression = On
This directive “zlib.output_compression” is used to transparently compress PHP output files using the zlib library of Linux. This can improve performance and especially time dial-up users should wait before seeing a PHP page. The default setting for this directive is "Off" and we change it for "On". It's important to note that the above "output_handler" directive must be empty if this directive is set to "On" as we do. This is a performance feature.

implicit_flush = Off
This directive “implicit_flush” is used to inform the output layer to flush itself automatically after every output block. This is useful for debugging purposes only and should never be used or set to "On" on production server since it has serious performance implications. This is a performance feature.

allow_call_time_pass_reference = Off
This directive “allow_call_time_pass_reference” is used to enable the ability to force PHP arguments to be passed by reference instead of by values at function call time. In future version of PHP, this method will be unsupported and all PHP arguments will be passed by values at function call time. This directive lets us choose which method we want to use for our PHP programs. By default, the setting for this directive is "On" but we are encouraged to try and turn this option "Off". This is what we do here but be sure that your scripts work properly when this option is set to "Off". If you have problem to make your scripts work when this option is set to "Off", then turn it back to "On".

safe_mode = On
This directive “safe_mode” is one of the most important setting in the php.ini file and the one that pose most problems for all of us. It has been made to solve the shared-server security problem that we can see when using Apache with PHP. When “safe_mode” is set to “On”, PHP checks to see if the owner of the current PHP script matches the owner of the file to be operated on by a file function. This means that every file related to the function implemented inside the PHP script file should have the same permission as the user that run the PHP script file or better, the user who own the PHP script file should have permission to run the file called by the function.

This is where problems appear when we set the value of this important directive to “On” because most advanced PHP software and especially those dealing with SQL databases provide internal PHP function that call external file on our Linux server and when the “safe_mode” directive is set to “On” those external PHP software do not have enough permission to access Linux files on our server because they run with the user permission that run the script, which is in general a user with less privileges on the server.

To solve this problem and to keep security of PHP as high as possible, we will play with the “safe_mode” directives by changing the default setting of “Off” to become “On” and will complete its parameters to make it work with the other directives directly associated with it as shown below. This is a security feature.

safe_mode_gid = Off
This directive “safe_mode_gid” is directly related to the above option (safe_mode). By default, Safe Mode when set to "On", does a UID compare check when opening files. If you want to relax this to a GID compare, then we can turn "On" the "safe_mode_gid" directive. Setting it to "On" perform the relaxed GID checking, setting it to "Off" (the default) performs UID checking. For optimum security, I recommend to keep the default value of "Off" here and only change it to "On" if you still have problem to run your PHP software on the server. This is a security feature.

safe_mode_include_dir = /var/lib/mysql
As we know now, when "safe_mode" is "On" and "safe_mode_gid" is "Off", UID/GID checks are made (this is what we want). The "safe_mode_include_dir" directive can be used to bypass this restriction. This is possible by specifying the path of the directory and/or subdirectory as a value of the directive. In our example, we define the path where our SQL database directory and subdirectories reside on our server. In this way UID/GID checks are bypassed when including files from this directory and its subdirectories and should make most PHP software workable with “safe_mode” enable on the web server supporting PHP. This is a security feature.

safe_mode_exec_dir =
When the "safe_mode" directive is set to "On", only executables located under the "safe_mode_exec_dir" directive line will be allowed to be executed via the exec family of functions. To complete our security with "safe_mode", we must list here any directories from where some executables reside for PHP to allow then to run on the server. In general and with databases connectivity, there are no executables to run, therefore, our value here is empty. If you have some special executables for your PHP software to run, then list here the complete path to the directory in question. This is a security feature.

;open_basedir =
This directive "open_basedir" is used to limits all file operations to the defined directory and below when "safe_mode" is set to "On". This directive makes most sense if used in a per-directory or per-virtualhost web server configuration file. In our configuration, we don't use it and this is why its parameter line is commented in our configuration file. This is a security feature.

safe_mode_allowed_env_vars = PHP_
This directive "safe_mode_allowed_env_vars" is used to define environment variables whose names begin with the prefixes supplied here. This directive contains a comma-delimited list of prefixes. In Safe Mode, the user may only alter environment variables whose names begin with the prefixes supplied here. With the default setting of this directive, users will only be able to set environment variables that begin with PHP_ (e.g. PHP_FOO=BAR). This is a security feature.

safe_mode_protected_env_vars = LD_LIBRARY_PATH
This directive "safe_mode_protected_env_vars" is used to define list of environment variables that the end user won't be able to change using putenv(). These variables will be protected even if the "safe_mode_allowed_env_vars" directive is set to allow changing them. The default setting is ok for most of us. This is a security feature.

disable_functions =
This directive "disable_functions" is used to disable individual functions for security reasons. It's important to note that this directive is NOT affected by whether Safe Mode is turned On or Off. If you know about some PHP function that you want to disable, then list them here. This is a security feature.

expose_php = Off
This directive "expose_php" is used to define whether PHP may expose the fact that it is installed on the server by adding its signature to the web server header. The default setting of "On" allow everyone form the external to determine whether we use PHP on our server or not. To disable this feature, you should set the value to "Off" (recommended). This is a security feature.

display_errors = Off
This directive "display_errors" is used to print out PHP errors as a part of the output. It's strongly encouraged to turn this feature "Off" to avoid revealing security information to end users, such as file paths on your web server, your database schema or other information. This is a security feature.

log_errors = On
This directive "log_errors" complements the above one. Any errors that occur during the execution of your script will be logged to your server's error log file. Along with setting the "display_errors" directive to "Off", this setup gives you the ability to fully understand what may have gone wrong, without exposing any sensitive information to remote users. This is a security feature.

register_globals = Off
One interesting feature of PHP that can be used to enhance security is configuring PHP with the "register_globals" directive set to "Off". By turning off the ability for any user-submitted variable to be injected into PHP code, you can reduce the amount of variable poisoning a potential attacker may inflict. They will have to take the additional time to forge submissions, and your internal variables are effectively isolated from user submitted data. Unfortunately some PHP software still uses this directive and if we set this option to "Off" as we do here, then something may break. I recommend you to set it to "Off" and test if your PHP software work with it. If you see that you have problem to make your application work when this setting is set to "Off", then change it to "On". This is a security and performance feature.

register_argc_argv = On
This directive "register_argc_argv" is used to tell PHP whether to declare the argv&argc variables (that would contain the GET information). If you don't use these variables, you should turn it "Off" for increased performance (recommended). Please note that some PHP software still required these variables to properly work on the server; this is why we keep the default value of "On" here. This is a performance feature.

magic_quotes_gpc = Off
This directive "magic_quotes_gpc" is used to define the "magic_quotes" state for GPC (**G**et/**P**ost/**C**ookie) operations on the web server. With the latest release of PHP, input data is no longer escaped with slashes so that it can be sent into SQL databases without further manipulation. This is a performance feature.

enable_dl = Off
This directive "enable_dl" is used to define whether or not to enable the dl() function (**d**ynamic **l**oading of PHP extensions). For security reason, you should turn dynamic loading "Off" because with dynamic loading set to "On", it's possible to ignore all the "safe_mode" and "open_basedir" security restrictions on the server. This is a security feature.

file_uploads = Off
This directive "file_uploads" is used to define whether you want to allow HTTP file uploads on the server or not. For security reason, I recommend you to disable this option by saying "Off" here. Remember what happen on the Internet when this setting was set to "On". Therefore enable at your own risk. This is a security feature.

```
;extension = imap.so
;extension = ldap.so
;extension = mysql.so
;extension = pgsql.so
```
This directive "extension" is used to enable specific applications with our PHP software. Remember that we have compiled PHP with support for IMAP, LDAP, MySQL, and PostgreSQL as loadable modules on the system. Therefore if we want to enable for example support for IMAP, we will simply need to uncomment its related line into the php.ini configuration file for PHP to know about it. In this way, we can enable or disable as we want any compiled in modules as listed above into our PHP with Apache. By default, I've disabled all available extension into the configuration file; therefore don't forget to uncomment any lines you want to enable support for.

```
[MySQL]
mysql.allow_persistent          = Off
mysql.max_persistent            = -1
mysql.max_links                 = -1
mysql.default_port              =
mysql.default_socket            = /var/lib/mysql/mysql.sock
mysql.default_host              =
mysql.default_user              =
mysql.default_password          =
```

For PHP scripts, the most expensive bottleneck is normally the CPU. Twin CPUs are probably more useful than two Gigabytes of RAM. When, using database connectivity with your PHP software, we were able to gain some important performance by switching to non-persistent database connections into the php.ini file. An alternative solution would have been to increase the MySQL "max_connections" parameter.

Step 2
As for any other modules with Apache, we have to activate the "libphp4.so" module for the web server to support it. This is possible by uncomment the line related to the module in question inside the httpd.conf file.

- Edit the **httpd.conf** file (vi /etc/httpd/httpd.conf), and change the line:

  ```
  #LoadModule php4_module         lib/apache/libphp4.so
  ```

 To read:

  ```
  LoadModule php4_module         lib/apache/libphp4.so
  ```

Step 3
Once the above line has been included/uncommented into the httpd.conf file of Apache to enable PHP feature, you must restart Apache for the changes to take effect.

- To restart Apache, use the following command:

  ```
  [root@deep /]# /etc/init.d/httpd restart
  Shutting down Apache:          [OK]
  Starting Apache:               [OK]
  ```

Running PHP in a chroot jail

This part is required only if you want to run PHP in a chroot jail mode. For this to work, Apache should have already been set to run in chroot jail environment. If this is not the case, then you should read the section related to how to make Apache run in a chroot jail before going into this part of running PHP in chroot environment.

Step 1
Running PHP in a chroot jail environment is really not difficult to accomplish, all we have to do, is to move all files and directories related to PHP into the Apache chroot jail for the web server to find the required PHP files to make it run.

- This can be done with the following commands:

```
[root@deep /]# mv /usr/lib/apache/libphp4.so /chroot/httpd/usr/lib/apache
[root@deep /]# mv /usr/lib/php4 /chroot/httpd/usr/lib/
[root@deep /]# mv /usr/share/pear /chroot/httpd/usr/share/
[root@deep /]# mv /etc/httpd/php.ini /chroot/httpd/etc/httpd/
```

Step 2
Finally, you must restart the Apache web server for the changes to take effect.

- To restart Apache, use the following command:

```
[root@deep /]# /etc/init.d/httpd restart
Shutting down Apache:          [OK]
Starting Apache:               [OK]
```

Running PHP with the PHP Accelerator program

The PHP Accelerator Cache is a PHP Zend engine extension capable of delivering a substantial acceleration of PHP scripts without requiring any script changes or loss of dynamic content but be warned that some scripts actually slow down when PHP Accelerator is installed. The consensus is that PHP Accelerator is good when your code has lots of loops.

If you intended to use this free program, you can download it from the PHP Accelerator website and place its library file into your system after expanding the archive.

These installation instructions assume
Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, at personal discretion).
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest PHP Accelerator version number is 1.3.1pre3

The procedures given in this chapter are likely to work on all Linux platforms, but we have only tested it on OpenNA Linux and Red Hat Linux.

Packages
The following is based on information listed by PHP Accelerator as of 2002/05/13. Please regularly check http://www.php-accelerator.co.uk/ for the latest status. We chose to install the required component from a source file because it provides the facility to fine tune the installation.

Source code is available from:
PHP Accelerator Homepage: http://www.php-accelerator.co.uk/
You must be sure to download: php_accelerator-1.3.1pre3_php-4.2.1_linux-glibc2.2.4.tgz

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- This can be done with the following commands:
```
[root@deep /]# cp php_accelerator-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf php_accelerator-version.tar.gz
```

Step 2
After that, move into the newly created PHP Accelerator directory and copy the file called php_accelerator_1.3.1pre3.so under /usr/lib/php4 directory.

- To copy the library file to your /usr/lib/php4 directory use the following commands:
```
[root@deep tmp]# cd php_accelerator-1.3.1pre3_php-4.2.1_linux-glibc2.2.4/
[root@deep php_accelerator...]# cp php_accelerator_1.3.1pre3.so /usr/lib/php4/
```

Step 3
Now, edit your **php.ini** file (vi /etc/httpd/php.ini) and add the following two lines near the extensions section of the configuration file.

```
; PHP Accelerator extension
zend_extension="/usr/lib/php4/php_accelerator_1.3.1pre3.so "
phpa.registration_key = 28ccb4e0144fa5f409fbfb3834b47358
```

NOTE: PHP Accelerator must be activated by setting a registration key for each domain (Actually each unique ServerName) to be enabled. Please see on the PHP Accelerator web site for more information about how to get your registration key. This key is what you should enter into the above line.

Step 4
Finally, you must restart the Apache web server for the changes to take effect.

- To restart Apache, use the following command:
```
[root@deep /]# /etc/init.d/httpd restart
Shutting down Apache:         [OK]
Starting Apache:              [OK]
```

Step 5
Now, to verify if the PHP Accelerator is running create a debug.php file under your root directory and access it with your web browser.

- Create the **debug.php** file (touch /home/httpd/html/debug.php) and add the following line inside the file.

```
<? echo phpinfo(); ?>
```

Step 6
Access the file with your web browser at http://www.domain.com/debug.php. The part of the output where the Zend Optimizer is listed will look something like this:

```
This program makes use of the Zend Scripting Language Engine:
Zend Engine v1.2.0, Copyright (c) 1998-2002 Zend Technologies
with the PHP Accelerator v1.3.1pre3, Copyright (c) 2001-2002, by Nick Lindridge
```

The <www.domain.com> is the address where your Apache web server lives, and <debug.php> is the PHP document we have created earlier to display the information and configuration of our Linux web server with PHP4 and PHP Accelerator support.

CHAPTER 45

Mod_Perl

IN THIS CHAPTER

1. Compiling - Optimizing & Installing `Mod_Perl`
2. Configuring `Mod_Perl`
3. Running `Mod_Perl` in a chroot jail

Linux Mod_Perl

Abstract

This chapter is another chapter related to the Apache web server, you should read it only if you have installed Apache on your system and want to provide and make it run with some additional feature. Here we talk about Mod_Perl with Apache.

Mod_Perl is used to directly incorporate a Perl interpreter into the Apache web server, so that the Apache web server can directly execute Perl code for better performance when running Perl programs. It's able to do it by linking the Perl runtime library into the Apache web server and provides an object-oriented Perl interface for Apache's C language API. The end result is a quicker CGI script turnaround process, since no external Perl interpreter has to be started by the web server.

It's a common misunderstanding to think that Mod_Perl is just a CGI replacement for Perl scripts into Apache, this is only a small part implemented by the Apache::Registry module. Apache modules written in Mod_Perl can do just about anything that Apache modules written in C can. You should install Mod_Perl only if you're installing the Apache web server and you'd like for it to directly incorporate a Perl interpreter. This could be useful if you have many CGI or Perl scripts available under your cgi-bin directory on your web server. Installing Mod_Perl will let you run all your existing CGI and Perl programs much faster without any modification of your codes.

As for the PHP **H**ypertext **P**reprocessor language, I've decided to explain you how to compile and install Mod_Perl with Apache as a module program that you may enable or disable as you like. This simply let us have more flexibility on our web server because we can upgrade the software without the need to rebuild the entire web server.

Finally before going into compilation, installation and configuration of the software, I would like to inform you that Mod_Perl software has been specially made to run with Apache 2.x and it's considered (at this time) experimental again. This means that newer version of the software fixing may bugs and more Apache 2 capable will certainly be available when you will read this chapter.

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, at personal discretion).
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest Mod_Perl version number is 1.99_04

The procedures given in this chapter are likely to work on all Linux platforms, but we have only tested it on OpenNA Linux and Red Hat Linux.

Packages

The following is based on information listed by Mod_Perl as of 2002/06/21. Please regularly check http://perl.apache.org/ for the latest status. We chose to install the required component from a source file because it provides the facility to fine tune the installation.

Source code is available from:
Mod_Perl Homepage: http://perl.apache.org/
You must be sure to download: mod_perl-1.99_04.tar.gz

Prerequisites

Mod_Perl requires that the software below is already installed on your system to be able to compile successfully. If this is not the case, you must install it. Please make sure you have this program installed on your machine before you proceed with this chapter.

- ✓ Apache is required to be able to use Mod_Perl in your system.
- ✓ Perl is required to be able to use Mod_Perl in your system.

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all the files installed on the system in the eventuality of an update in the future. To solve the problem, it is a good idea to make a list of files on the system before you install Mod_Perl, and then one afterwards, and then compare them using the **diff** utility to find out what files were placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep root]# find /* > Mod_Perl1
  ```
- And the following one after you install the software:
  ```
  [root@deep root]# find /* > Mod_Perl2
  ```
- Then use the following command to get a list of what changed:
  ```
  [root@deep root]# diff Mod_Perl1 Mod_Perl2 > Mod_Perl-Installed
  ```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. In the example above, we use the /root directory of the system to store all generated list files.

Compiling - Optimizing & Installing Mod_Perl

Below are the steps that you must make to configure, compile and optimize the Mod_Perl software before installing it onto your system. First off, we install the program as the user "root" so as to avoid permissioning problems.

Step 1

Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- This can be done with the following commands:

```
[root@deep /]# cp mod_perl-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf mod_perl-version.tar.gz
```

Step 2

After that, move into the newly created Mod_Perl source directory and perform the following steps to configure and optimize Mod_Perl for your system.

- To move into the newly created Mod_Perl source directory use the command:

```
[root@deep tmp]# cd mod_perl-1.99_04/
```

Step 3

There is one Mod_Perl source file to modify before going in configuration and compilation of the program; the change allow us to fix a problem related to the location where we've installed the Apache modules and headers files on our system. Mod_Perl need to know where it should look for these files and we have to inform it about the right locations here.

- Edit the **Build.pm** file (vi +49 lib/Apache/Build.pm) and change the lines:

```
INCLUDEDIR => 'include',
LIBEXECDIR => 'modules',
```

To read:

```
INCLUDEDIR => 'apache',
LIBEXECDIR => 'apache',
```

Step 4

Once the required modification has been made into the related source file of Mod_Perl, it is time to configure, optimize and install it on our system. We produce a list of files on the system before we install the software and one afterwards then compare them using the **diff** utility to find out what files were placed where and finally install Mod_Perl.

```
[root@deep mod_perl-1.99_04]# perl Makefile.PL MP_AP_PREFIX=/usr/include
[root@deep mod_perl-1.99_04]# make
[root@deep mod_perl-1.99_04]# cd
[root@deep root]# find /* > Mod_Perl1
[root@deep root]# cd /var/tmp/mod_perl-1.99_04/
[root@deep mod_perl-1.99_04]# make install
[root@deep mod_perl-1.99_04]# cd
[root@deep root]# find /* > Mod_Perl2
[root@deep root]# diff Mod_Perl1 Mod_Perl2 > Mod_Perl-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

Step 5
Once the compilation, optimization and installation of the software has completed, we can free up some disk space by deleting the program tar archive and the related source directory, since they are no longer needed.

- To delete Mod_Perl and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf mod_perl-version/
[root@deep tmp]# rm -f mod_perl-version.tar.gz
```

The rm command as used above will remove all the source files we have used to compile and install Mod_Perl. It will also remove the Mod_Perl compressed archive from the /var/tmp directory.

Configuring Mod_Perl

After Mod_Perl has been built and installed successfully on your system, the next step is to configure and customize its configuration file to fit your needs. Mod_Perl do not have any configuration file to configure, it is just a module program that you must enable into your web server configuration file to be able to use it. Nothing else is required.

Step 1
As for any other modules with Apache, we have to activate the "mod_perl.so" module for the web server to support it. This is possible by uncomment the line related to the module in question inside the httpd.conf file.

- Edit the **httpd.conf** file (vi /etc/httpd/httpd.conf), and change the line:

```
#LoadModule perl_module          lib/apache/mod_perl.so
```

To read:

```
LoadModule perl_module          lib/apache/mod_perl.so
```

Step 2
Once the above line has been included/uncommented into the httpd.conf file of Apache to enable Mod_Perl feature, you must restart Apache for the changes to take effect.

- To restart Apache, use the following command:

```
[root@deep /]# /etc/init.d/httpd restart
Shutting down Apache:         [OK]
Starting Apache:              [OK]
```

Running Mod_Perl in a chroot jail

This part is required only if you want to run Mod_Perl in chroot jail mode. For this to work, Apache should have already been set to run in a chroot jail environment. If this is not the case, then you should read the section related to how to make Apache run in chroot jail before going into this part of running Mod_Perl in chroot environment.

Step 1
Running Mod_Perl in a chroot jail environment is really not difficult to accomplish, all we have to do, is to move all files and directories related to Mod_Perl into the Apache chroot jail for the web server to find the required Mod_Perl files to make it run. This means that we have to make a copy of the whole /usr/lib/perl5 directory and binary into our chroot jail structure because as you can guest, Mod_Perl required Perl language and related files to work.

- This can be done with the following commands:

```
[root@deep /]# mkdir -p /chroot/httpd/usr/bin
[root@deep /]# cp /usr/bin/perl /chroot/httpd/usr/bin/
[root@deep /]# cp -a /usr/lib/perl5 /chroot/httpd/usr/lib/
[root@deep /]# cd /usr/lib/apache/
[root@deep apache]# cp mod_perl.so /chroot/httpd/usr/lib/apache/
```

Step 2
Finally, you must restart the Apache web server for the changes to take effect.

- To restart Apache, use the following command:

```
[root@deep /]# /etc/init.d/httpd restart
Shutting down Apache:          [OK]
Starting Apache:               [OK]
```

Further documentation

For more details, there are some manual pages about Mod_Perl that you could read:

```
$ man APR::Table (3)            - A Perl API for manipulating opaque string-content table.
$ man Apache::Build (3)         - Methods for locating and parsing bits of Apache source code.
$ man Apache::RequestRec (3)    - A Perl API for Apache request object.
$ man ModPerl::Code (3)         - Generate mod_perl glue code.
```


Part XV
Net Bios
Protocol

CHAPTER 46

Samba

IN THIS CHAPTER

1. **Compiling - Optimizing & Installing `Samba`**
2. **Configuring `Samba`**
3. **Running `Samba` with `TLS/SSL` support**
4. **Securing `Samba`**
5. **Optimizing `Samba`**
6. **`Samba` Administrative Tools**
7. **`Samba` Users Tools**

Linux Samba

Abstract

Enterprise-level organizations often handle many kinds of different operating systems, and have the need to keep them in a networked environment for files sharing and printers. Employees may work on workstations like Linux, Microsoft Windows 95/98/2000/NT/XP, OS/2 or Novel, and still need to access the server in their daily work. A Linux server with Samba support can be used to respond for these kinds of activities.

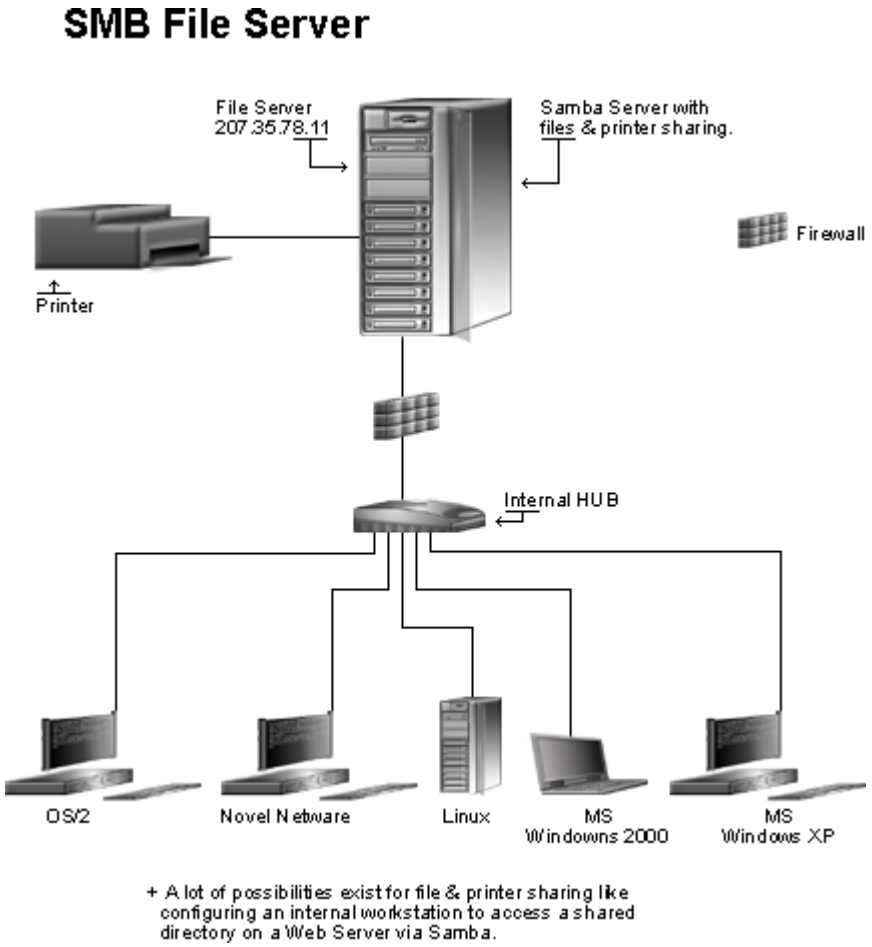
Samba is a strong network service for file and print sharing that works on the majority of operating systems available today. When well implemented by the administrator, it's faster and more secure than the native file sharing services available on Microsoft Windows machines.

Samba is the protocol by which a lot of PC-related machines share files and printers, and other information, such as lists of available files and printers. Operating systems that support this natively include Windows 95/98/2000/NT/XP, OS/2, and Linux, and add on packages that achieve the same thing are available for DOS, Windows, VMS, Unix of all kinds, MVS, and more.

Apple Macs and some Web Browsers can speak this protocol as well. Alternatives to SMB include Netware, NFS, AppleTalk, Banyan Vines, Decnet etc; many of these have advantages but none are both public specifications and widely implemented in desktop machines by default.

Samba software includes an SMB server, to provide Windows NT and LAN Manager-style file and print services to SMB clients such as Windows 2000, Warp Server, smbfs and others, a Net BIOS (rfc1001/1002) name server, which amongst other things gives browsing support, an ftp-like SMB client so that you can access PC resources (disks and printers) from Unix, Netware and other operating systems, and finally, a tar extension to the client for backing up PCs.

In this chapter, we will explain and cover some of the basic ways in which you can adjust the configuration to improve the server's performance. Also, for the interested users, we'll provide a procedure to run Samba with SSL protocol support. Running Samba with SSL support will work perfectly for Unix-to-Unix platforms but not for Windows to Unix. This is in particularly due to the fact that at this time Microsoft has not reviewed its File Sharing system on Windows.


SMB File Server
File Server
207.35.78.11
Samba Server with
files & printer sharing.
Firewall
Printer
Internal HUB
OS/2
Novel Netware
Linux
MS
Windowns 2000
MS
Windows XP
+ A lot of possibilities exist for file & printer sharing like
configuring an internal workstation to access a shared
directory on a Web Server via Samba.

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, at personal discretion).
Installations were tested on OpenNA Linux & Red Hat Linux 7.3.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest Samba version number is 2.2.5

The procedures given in this chapter are likely to work on all Linux platforms, but we have only tested it on OpenNA Linux and Red Hat Linux.

Packages

The following is based on information listed by Samba as of 2002/06/19. Please regularly check http://www.samba.org/ for the latest status. We chose to install the required component from a source file because it provides the facility to fine tune the installation.

Source code is available from:
Samba Homepage: http://www.samba.org/
Samba FTP Site: 198.186.203.85
You must be sure to download: samba-2.2.5.tar.gz

Prerequisites

Samba requires that the software below is already installed on your system to be able to compile successfully. If this is not the case, you must install it. Please make sure you have this program installed on your machine before you proceed with this chapter.

- ✓ OpenSSL is required to be able to use Samba with SSL support in your system.

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all the files installed on the system in the eventuality of an update in the future. To solve the problem, it is a good idea to make a list of files on the system before you install Samba, and then one afterwards, and then compare them using the **diff** utility to find out what files were placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep root]# find /* > Samba1
  ```
- And the following one after you install the software:
  ```
  [root@deep root]# find /* > Samba2
  ```
- Then use the following command to get a list of what changed:
  ```
  [root@deep root]# diff Samba1 Samba2 > Samba-Installed
  ```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. In the example above, we use the /root directory of the system to store all generated list files.

Compiling - Optimizing & Installing Samba

Below are the steps that you must make to configure, compile and optimize the samba software before installing it onto your system. First off, we install the program as the user "root" so as to avoid permissioning problems.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- This can be done with the following commands:
```
[root@deep /]# cp samba-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf samba-version.tar.gz
```

Step 2
In order to check that the version of Samba, which you are going to install, is an original and unmodified one, use the command described below to check its MD5 hashes checksum.

- To verify the MD5 checksum of Samba, use the following command:
```
[root@deep tmp]# md5sum samba-2.2.5.tar.gz
```

This should yield an output similar to this:
```
4899dfdca88d86e7809c16f23c24eecc  samba-2.2.5.tar.gz
```

Now check that this checksum is exactly the same as the one available into a file called "samba-2.2.5.tar.gz.md5" on the Samba FTP site: 198.186.203.85

Step 3
After that, move into the newly created Samba source subdirectory called "source" and perform the following steps before configuring and optimizing Samba for your system.

- To move into the newly created Samba source subdirectory use the command:
```
[root@deep tmp]# cd samba-2.2.5/source/
```

Step 4
There are some source files to modify before going in configuration and compilation of the program; the changes allow us to fix location of installed files in our operating system and tool to use to compile one Samba utility. The first modification that we do is to relocate the lib directory of Samba to be under the /usr/bin directory.

- Edit the **smbsh.in** file (vi +3 smbwrapper/smbsh.in) and change the line:

  ```
  SMBW_LIBDIR=${SMBW_LIBDIR-@builddir@/smbwrapper}
  ```

 To read:

  ```
  SMBW_LIBDIR=${SMBW_LIBDIR-/usr/bin}
  ```

Step 5
Here we specify to use the GNU Linux version of the awk text processing utility instead of the Bell Labs research version of awk program for the "smbpasswd" file to compile successfully.

- Edit the **convert_smbpasswd** file (vi +10 script/convert_smbpasswd) and change the line:

  ```
  nawk 'BEGIN {FS=":"}
  ```

 To read:

  ```
  gawk 'BEGIN {FS=":"}
  ```

Step 6
Once the required modifications have been made into the related source files of Samba as explained previously, it is time configure and optimize it for our system.

- To configure and optimize Samba use the following compilation lines:

```
perl -pi -e "s|-symbolic||" Makefile.in
CFLAGS="-O2 -march=i686 -funroll-loops -D_GNU_SOURCE"; export CFLAGS
./configure \
--prefix=/usr \
--sysconfdir=/etc \
--localstatedir=/var \
--libdir=/etc/samba \
--mandir=/usr/share/man \
--with-privatedir=/etc/samba \
--with-lockdir=/var/lock/samba \
--with-piddir=/var/run/samba \
--with-swatdir=/usr/share/swat \
--with-codepagedir=/usr/share/samba/codepages \
--with-sslinc=/usr/include \
--with-ssllib=/usr/lib \
--with-ssl \
--with-fhs \
--with-pam \
--with-syslog \
--with-quotas \
--with-utmp
```

Step 7
Now, we must make a list of all existing files on the system before installing the software and one afterwards then compare them using the **diff** utility tool of Linux to find out what files are placed where and finally install Samba on the server.

```
[root@deep source]# make
[root@deep source]# cd
[root@deep root]# find /* > Samba1
[root@deep root]# cd /var/tmp/samba-2.2.5/source/
[root@deep source]# make install
[root@deep source]# install -m0511 script/mksmbpasswd.sh  /usr/bin/
[root@deep source]# rm -rf /usr/private/
[root@deep source]# rm -rf /usr/share/swat/
[root@deep source]# rm -f /usr/sbin/swat
[root@deep source]# rm -f /usr/share/man/man8/swat.8
[root@deep source]# mkdir -m0755 /var/lock/samba
[root@deep source]# mkdir -m1777 /var/spool/samba
[root@deep source]# mkdir -m0700 /var/log/samba
[root@deep source]# strip /usr/sbin/smbd
[root@deep source]# strip /usr/sbin/nmbd
[root@deep source]# cd
[root@deep root]# find /* > Samba2
[root@deep root]# diff Samba1 Samba2 > Samba-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

Step 8
Once the compilation, optimization and installation of the software has completed, we can free up some disk space by deleting the program tar archive and the related source directory, since they are no longer needed.

- To delete Samba and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf samba-version/
[root@deep tmp]# rm -f samba-version.tgz
```

The rm command as used above will remove all the source files we have used to compile and install Samba. It will also remove the Samba compressed archive from the /var/tmp directory.

Configuring Samba

After Samba has been built and installed successfully on your system, the next step is to configure and customize its configuration files to fit your needs.

- ✓ /etc/samba/smb.conf: (The Samba Configuration File)
- ✓ /etc/samba/lmhosts: (The Samba Net BIOS Mapping File)
- ✓ /etc/sysconfig/samba: (The Samba System Configuration File)
- ✓ /etc/pam.d/samba: (The Samba PAM Support Configuration File)
- ✓ /etc/logrotate.d/samba: (The Samba Log Rotation File)
- ✓ /etc/init.d/smb: (The Samba Initialization File)

/etc/samba/smb.conf: The Samba Configuration File

The /etc/samba/smb.conf file is the main configuration file for the Samba suite and contains runtime configuration information, in which you can specify directories you want to access from Windows client machines, IP addresses that are authorized to connect, how the File Sharing Server must run as, and so on through entries consisting of sections and parameters.

There are three special sections available with Samba. The first section called [global] contains global configuration directives common to all shares and become the defaults for sections, which do not specifically define certain items (unless they are over-ridden on a per-share basis).

The second section called [homes] allows services connecting clients to their home directory to be created on the fly by the File Sharing Server. This special section can represent any account on the machine, which isn't always desirable. For example, it can potentially create a share for root, bin, sys, and the like users. Therefore to eliminate this potential risk we must set an invalid users option in the [homes] section to protect against this.

The last section called [printers] works like the [homes] section but for printers. It allows users to connect to any printer specified in the configuration file.

A lot of options exist, and it's important to read the documentation that comes with Samba for more information on each of different settings and parameters available.

The following configuration is a full working configuration file for Samba with encrypted password support. Also, it's important to note that I comment in this Samba configuration only parameters that relate to security and optimization, and leave all others to your own research.

In the example below, I have created just one directory called [tmp], and have allowed only class C machine IP address ranges to connect on the Samba server to this directory. Therefore don't forget to add your own directories from which you want your client machines to connect. Text in bold is the parts of the configuration file that must be customized and adjusted to satisfy your needs.

- Create the **smb.conf** file (touch /etc/samba/smb.conf) and add the following lines:

```
[global]

        workgroup = OPENNA
        server string = OpenNA Samba Server
        encrypt passwords = True
        security = user
        smb passwd file = /etc/samba/smbpasswd
        log file = /var/log/samba/log.%m
        max log size = 0
        socket options = IPTOS_LOWDELAY TCP_NODELAY
        deadtime = 15
        getwd cache = Yes
        lpq cache time = 45
        domain master = Yes
        local master = Yes
        preferred master = Yes
        os level = 65
        dns proxy = Yes
        wins support = Yes
        name resolve order = wins lmhosts host bcast
        bind interfaces only = True
```

```
        interfaces = eth0 192.168.1.1/24 127.0.0.1
        hosts deny = ALL
        hosts allow = 192.168.1. 207.35.78. 127.0.0.1
        debug level = 1
        create mask = 0644
        directory mask = 0755
        oplocks = True
        level2 oplocks = True
        read raw = No
        write cache size = 262144

[homes]
        comment = Home Directories
        browseable = No
        read only = Yes
        invalid users = root bin daemon sync nobody sys tty disk mem kmem

[printers]
        comment = Remote Printers
        path = /var/spool/samba
        browseable = No
        printable = Yes
        invalid users = root bin daemon sync nobody sys tty disk mem kmem

[tmp]
        comment = Temporary File Space
        path = /tmp
        read only = No
        valid users = smbadmin
        invalid users = root bin daemon sync nobody sys tty disk mem kmem
```

This tells the `smb.conf` file to set itself up for this particular configuration setup with:

`[global]`

`workgroup = OPENNA`
This parameter "`workgroup`" specifies the workgroup your server will appear to be in when queried by clients. It's important to have the same workgroup name on both clients and servers machines. Therefore don't forget to set the same workgroup name in the client part from which you want to connect to the server.

`server string = OpenNA Samba Server`
This parameter "`server string`" specifies the string that you wish to show to your users in the printer comment box in print manager, or to the `IPC` connection when using the "`net view`" command under Windows machines. It can be any string that you wish to show to your users.

`encrypt passwords = True`
This parameter "`encrypt passwords`" if set to "`True`" instructs `Samba` to use encrypted passwords instead of plain text password when negotiating with the client. Sniffer program will not be able to detect your password when it is encrypted. This option always must be set to "`True`" for security reasons. This is a security feature.

`security = user`
This parameter "`security`", if set to "`user`", specifies that a client must first "log-on" with a valid username and password, or the connection will be refused. This means that a valid username and password for the client must exist in your `/etc/passwd` file on the Linux server and in the `/etc/smbpasswd` file of the `Samba` server, or the connection from the client will fail. See "Securing Samba" in this chapter for more information about the "`smbpasswd`" file. This parameter is one of the most important settings in the `smb.conf` file. This is a security feature.

`smb passwd file = /etc/samba/smbpasswd`
This parameter "`smb passwd file`" specifies the path to the encrypted "`smbpasswd`" file. The "`smbpasswd`" file is a copy of the `/etc/passwd` file of the Linux system containing valid usernames and passwords of clients allowed to connect to the `Samba` server. The `Samba` software reads this file (`smbpasswd`) when a connection is requested.

`log file = /var/log/samba/log.%m`
This parameter "`log file`" specifies the locations and names of `Samba` log files. With the name extension "`%m`", it allows you to have separate log files for each different user or machine that logs on your `Samba` server.

`socket options = IPTOS_LOWDELAY TCP_NODELAY`
This parameter "`socket options`" specifies parameters that you can include in your `smb.conf` configuration file to tune and improve your `Samba` server for optimal performance. By default we chose to tune the connection for a local network, and improve the performance of the `Samba` server for transferring files. This is a performance feature.

`deadtime = 15`
This parameter "`deadtime`" specifies the number of minutes to wait for client inactivity before considering that the connection is dead, and close it. A `deadtime` of zero (the default setting) indicates that no auto-disconnection should be performed. Using this parameter with a timeout of a few minutes is recommended for better performance of the systems. This is a performance feature.

`getwd cache = Yes`
This parameter "`getwd cache`" if set to "`Yes`" specifies to reduce the time taken for `getwd()` calls by using a caching algorithm. This is a performance feature.

`lpq cache time = 45`
This parameter "`lpq cache time`" specifies how long `lpq` info will be cached on memory to prevent the `lpq` command being called too often. A large value is recommended when your `lpq` command is very slow on the system. This is a performance feature.

`domain master = Yes`
This parameter "`domain master`" specifies to set "`nmbd`", which is the `Net BIOS` name server daemon, as a domain master browser for its given workgroup and enables WAN-wide browse list collation. This option usually must be set to "`Yes`" only on ONE `Samba` server for all OTHER `Samba` servers on the same network and workgroup.

local master = Yes
This parameter "local master" allows "nmbd", which is the Net BIOS name server daemon, to try to become a local master browser on a subnet. Like the above, usually this option must be set to "Yes" only on ONE Samba server that acts as a local master on a subnet for all the OTHER Samba servers on your network. Setting this parameter to "Yes", doesn't guaranty that Samba will become the local master browser on a subnet, it just ensure that Samba will participate in elections for local master browser. Use it in conjunction with parameters "domain master", and "preferred master" below.

preferred master = Yes
This parameter "preferred master" specifies and controls if "nmbd" is a preferred master browser for its workgroup. Once again, this must usually be set to "Yes" on ONE server for all the others on your network. Use it in conjunction with parameters "domain master", and "local master".

os level = 65
This parameter "os level" specifies by its integer value whether "nmbd" has a chance of becoming a local master browser for the Workgroup in the local broadcast area. The number 65 will win against any NT Server. If you have an NT Server on your network, and want to set your Linux Samba server to be a local master browser for the Workgroup in the local broadcast area then you must set the "os level" option to 65. Also, this option must be set only on ONE Linux Samba server, and must be disabled on all other Linux Samba servers you may have on your network. Use it in conjunction with parameters "domain master", "local master", and "preferred master".

dns proxy = Yes
This parameter "dns proxy" if set to "Yes" specifies that "nmbd" when acting as a WINS server and finding that a Net BIOS name has not been registered, should treat the Net BIOS name word-for-word as a DNS name and do a lookup with the DNS server for that name on behalf of the name-querying client. Configuring the Samba server to act as a WINS server is a good thing for its performance. I recommend using your Samba server that runs with parameters "domain master", "local master", "preferred master", and "os level" set to "Yes" with this option "dns proxy" set to "Yes" too for better performance of your system.

wins support = Yes
This parameter "wins support" if set to "Yes" specifies that "nmbd" on the system will act as a WINS server. For better performance, it is recommended to set at least one Samba server in your network to be a WINS server. Note that you should NEVER set this to "Yes" on more than one machine in your network. It is a good idea to set your Samba server that runs with parameters "domain master", "local master", "preferred master", and "os level" set to "Yes" to become the WINS server on the network (as we do here).

name resolve order = wins lmhosts host bcast
This parameter "name resolve order" specifies what naming services to use in order to resolve host names to IP addresses, and in what order. The parameters we chose causes the local "lmhosts" file of samba to be examined first, followed by the rest. This is a performance feature.

bind interfaces only = True
This parameter "bind interfaces only" if set to "True", allows you to limit what interfaces on the server will serve "SMB" requests. This is a security feature. The configuration parameter "interfaces" below completes this option.

`interfaces = eth0 192.168.1.1/24 127.0.0.1`
This parameter "`interfaces`" allows you to override the default network interface list that `Samba` will use for browsing, name registration and other `NBT` traffic. By default, `Samba` will query the kernel for the list of all active interfaces and use any interface that it will find. With the above option, `Samba` will only listen on interface "`eth0`" on the `IP` addresses `192.168.1.1/24` and `127.0.0.1`.

This is a security feature, and completes the above configuration parameter "`bind interfaces only`". Please note that if the network address `127.0.0.1` is not added to the "`interfaces`" parameter list then `smbpasswd` will fail to connect in its default mode since we use the "`bind interfaces only`" parameter in conjunction with the "`interfaces`" parameter here. Therefore don't forget to add `127.0.0.1` to the "`interfaces`" parameter list above.

`hosts deny = ALL`
This parameter "`hosts deny`" specifies the list of hosts that are NOT permitted access to `Samba` services unless the specific services have their own lists to override this one. For simplicity, we deny access to all hosts by default, and allow specific hosts in the "`hosts allow`" parameter list as shown below. This is a security feature.

`hosts allow = 192.168.1. 207.35.78. 127.0.0.1`
This parameter "`hosts allow`" specifies which hosts are permitted to access a `Samba` service. In our example we allow by default all hosts from IP class C `192.168.1.*`, `207.35.78.*` and our localhost `127.0.0.1` to access the `Samba` server. Note that the `localhost` must always be set or you will receive some error messages. This is a security feature.

`debug level = 1`
This parameter "`debug level`" allows the logging level to be specified in the "`smb.conf`" file. If you set the debug level higher than 2 then you may suffer a large drop in performance. This is because the server flushes the log file after each operation, which can be very expensive. This is a performance feature.

`create mask = 0644`
This parameter "`create mask`" specifies and sets the necessary permissions according to the mapping from DOS modes to UNIX permissions. With this option set to `0644`, all files copying or creating from a Windows system to the Unix system will have a permission of `0644` by default. This is a security feature.

`directory mask = 0755`
This parameter "`directory mask`" specifies and set the octal modes, which are used when converting DOS modes to UNIX modes when creating UNIX directories. With this option set to `0755`, all directories copying or creating from a Windows system to the Unix system will have a permission of `0755` by default. This is a security feature.

`oplocks = True`
This parameter "`oplocks`", tells `smbd` whether to issue `oplocks` (opportunistic locks) to file open requests. The `oplock` code can dramatically improve the speed of access to files on `Samba` servers and it is recommended to set this option to "`True`". This is a performance feature.

`level2 oplocks = True`
This parameter "`level2 oplocks`" if set to "`True`", will increase the performance for many accesses of files that are not commonly written (such as `.EXE` application files). It is important for the "`oplocks`" (opportunistic locks) parameter to be set to "`True`" on this share in order for the "`level2 oplocks` " parameter to have any effect. This is a performance feature.

`read raw = No`
This parameter “`read raw`” controls whether or not the server will support the raw read `SMB` requests when transferring data to clients. Note that memory mapping is not used by the "`read raw`" operation. Thus, you may find memory mapping is more effective if you disable "`read raw`" using "`read raw = No`", like we do. This is a performance feature.

`write cache size = 262144`
This parameter “`write cache size`” allows `Samba` to improve performance on systems where the disk subsystem is a bottleneck. The value of this option is specified in bytes, and a size of 262,144 represents a 256k-cache size per file. It is to yours to set this parameter adequately related to the size of files that you hope to share with your server. If the majority of sharing files are between 512K in size, you could set the parameter to “`524288`”. This is a performance feature.

`[tmp]`

`comment = Temporary File Space`
This parameter “`comment`” allow you to specify a comment that will appear next to a share when a client does queries to the server either via the network neighborhood or via "`net view`" to list what shares are available.

`path = /tmp`
This parameter “`path`” specifies a directory to which the user of the service is to be given access. In our example this is the “`tmp`” directory of the Linux server.

`read only = No`
This parameter “`read only`” specifies if users should be allowed to only read files or not. In our example, since this is a configuration for the “`tmp`” directory of the Linux server, users can do more than just read files.

`valid users = smbadmin`
This parameter “`valid users`” specifies a list of users that should be allowed to login to this service. In our example only the user “`smbadmin`” is allowed to access the service.

`invalid users = root bin daemon sync nobody sys tty disk mem kmem`
This parameter “`invalid users`” specifies a list of users that should not be allowed to login to this service. This is really a "paranoid" check to ensure an improper setting does not breach your security. It is recommended that you include all default users that run daemons on the server. This is a security feature.

/etc/samba/lmhosts: The Samba Net BIOS Mapping File

The "lmhosts" file is the Samba Net BIOS name to IP address mapping file. It is very similar to the /etc/hosts file format, except that the hostname component must correspond to the Net BIOS naming format. Text in bold is the parts of the script initialization file that must be customized and adjusted to satisfy your needs.

- Create the **lmhosts** file (touch /etc/samba/lmhosts) and add the following lines:

```
# Sample Samba lmhosts file.
#
127.0.0.1       localhost
192.168.1.30    station1
192.168.1.31    station2
```

In our example, this file contains three IP to Net BIOS name mappings. The localhost (127.0.0.1), which is always require, the client machine called station1 (192.168.1.30) and another client machine called station2 (192.168.1.31). Don't forget to list your entire client machine name in this file.

/etc/sysconfig/samba: The Samba System Configuration File

The /etc/sysconfig/samba file is used to specify Samba system configuration information, such as if additional options are required to be passed to smbd and nmbd daemons at startup.

- Create the **samba** file (touch /etc/sysconfig/samba) and add the following lines:

```
# Start SMBD as a daemon on the server.
SMBDOPTIONS="-D"

# Start NMBD as a daemon on the server.
NMBDOPTIONS="-D -H /etc/samba/lmhosts"

# Start WINBINDD as a daemon on the server.
WINBINDOPTIONS=""
```

The "SMBDOPTIONS" and "NMBDOPTIONS" parameters with the "-D" options instructs samba server to operate as a daemon on the system. These values must be specified in this file since by default, the server will NOT operate as a daemon. Operating the server as a daemon is the recommended way of running Samba in your server.

/etc/pam.d/samba: The Samba PAM Support Configuration File

For better security of Samba, we will configure it to use PAM password authentication support. To do that, you must create the /etc/pam.d/samba file and add the following parameters inside it.

- Create the **samba** file (touch /etc/pam.d/samba) and add the following lines:

```
auth       required     /lib/security/pam_stack.so service=system-auth
auth       required     /lib/security/pam_nologin.so
account    required     /lib/security/pam_stack.so service=system-auth
account    required     /lib/security/pam_access.so
account    required     /lib/security/pam_time.so
password   required     /lib/security/pam_stack.so service=system-auth
session    required     /lib/security/pam_stack.so service=system-auth
session    required     /lib/security/pam_limits.so
session    optional     /lib/security/pam_console.so
```

/etc/logrotate.d/samba: The Samba Log Rotation File

This file allows the Samba server to automatically rotate its log files at the specified time. Here we'll configure the /etc/logrotate.d/samba file to rotate each week its log files automatically.

- Create the **samba** file (touch /etc/logrotate.d/samba) and add the lines:

```
/var/log/samba/*.log {
    notifempty
    missingok
    sharedscripts
    copytruncate
    postrotate
        /bin/kill -HUP `cat /var/run/smbd.pid /var/run/nmbd.pid 2>
/dev/null` 2> /dev/null || true
    endscript
}
```

/etc/init.d/smb: The Samba Initialization File

The smb script file is responsible to automatically start and stop the Samba server on your Linux system. Loading the smbd and nmbd daemons as standalone daemons will eliminate load time and will even reduce swapping since non-library code will be shared.

Please note that the following script is suitable for Linux operating systems that use System V. If you Linux system use some other methods like BSD, you'll have to adjust the script bellow to make it work for you.

Step 1
Create the **smb** script file (touch /etc/init.d/smb) and add the following lines:

```
#!/bin/bash

# This shell script takes care of starting and stopping Samba.
#
# chkconfig: 345 91 35
# description: Starts and stops the Samba smbd and nmbd daemons \
#              used to provide SMB network services.
#
# config: /etc/samba/smb.conf
# pidfile: /var/run/samba/smbd.pid
# pidfile: /var/run/samba/nmbd.pid

# Source function library.
. /etc/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Source for additional options if we have them.
if [ -f /etc/sysconfig/samba ] ; then
        . /etc/sysconfig/samba
fi

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0
```

```
# Avoid using root's TMPDIR
unset TMPDIR

# If Samba is not available stop now.
[ -f /usr/sbin/smbd ] || exit 0
[ -f /usr/sbin/nmbd ] || exit 0

# Path to the Samba binary.
smbd=/usr/sbin/smbd
nmbd=/usr/sbin/nmbd

RETVAL=0
prog="Samba"

start() {
        KIND="SMB"
        echo -n $"Starting $prog $KIND: "
        daemon $smbd $SMBDOPTIONS
        RETVAL=$?
        echo
        KIND="NMB"
        echo -n $"Starting $prog $KIND: "
        daemon $nmbd $NMBDOPTIONS
        RETVAL2=$?
        echo
        [ $RETVAL -eq 0 -a $RETVAL2 -eq 0 ] && touch /var/lock/subsys/smb || \
           RETVAL=1
        return $RETVAL
}

stop() {
        KIND="SMB"
        echo -n $"Shutting down $prog $KIND: "
        killproc $smbd
        RETVAL=$?
        echo
        KIND="NMB"
        echo -n $"Shutting down $prog $KIND: "
        killproc $nmbd
        RETVAL2=$?
        echo
        [ $RETVAL -eq 0 -a $RETVAL2 -eq 0 ] && rm -f /var/lock/subsys/smb || \
          RETVAL=1
        return $RETVAL
}

# See how we were called.
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  status)
        status $smbd
        status $nmbd
        RETVAL=$?
        ;;
  restart)
        stop
        start
        RETVAL=$?
```

```
        ;;
  condrestart)
        if [ -f /var/lock/subsys/smb ]; then
                stop
                start
                RETVAL=$?
        fi
        ;;
  *)
        echo $"Usage: $0 {start|stop|status|restart|condrestart}"
        exit 1
esac
exit $RETVAL
```

Step 2
Once the smb script file has been created, it is important to make it executable, change its default permissions, create the necessary links and start it. Making this file executable will allow the system to run it, changing its default permission is to allow only the root user to change this file for security reason, and creation of the symbolic links will let the process control initialization of Linux which is in charge of starting all the normal and authorized processes that need to run at boot time on your system to start the program automatically for you at each reboot.

- To make this script executable and to change its default permissions, use the commands:
```
[root@deep /]# chmod 700 /etc/init.d/smb
[root@deep /]# chown 0.0 /etc/init.d/smb
```

- To create the symbolic rc.d links for Samba, use the following commands:
```
[root@deep /]# chkconfig --add smb
[root@deep /]# chkconfig --level 345 smb on
```

- To start Samba daemons manually, use the following command:
```
[root@deep /]# /etc/init.d/smb start
Starting Samba SMB:          [OK]
Starting Samba NMB:          [OK]
```

Running Samba with TLS/SSL support

This section applies only if you want to run Samba through SSL connection. Usually running Samba with SSL support is only required when you share files with the external through the Internet. For corporate network that runs Samba on an LAN for their Windows client machines, this is not useful since at this time Microsoft doesn't provide with their operating systems SSL support for File Sharing.

There is from my knowledge one program called "stunnel", which could help to solve this problem with Windows machines but I don't recommend you to use it. Unfortunately the best will be to wait and hope that Microsoft will provide SSL support with File Sharing in future upgrade of its operating systems. From now you can use this new feature of running Samba through SSL connection with operating systems like Linux with the use of its smbclient program.

Below I show you how to set up the required certificate to be able to use Samba through SSL connection. Again, the principle is exactly the same as for creating a certificate for a web server (refer to OpenSSL chapter if you have problem creating the certificates).

Step 1
First you have to know the **F**ully **Q**ualified **D**omain **N**ame (FQDN) of the File Sharing Server for which you want to request a certificate. When you want to access your File Sharing Server through smb.domain.com then the FQDN of your File Sharing Server is smb.domain.com.

Step 2
Second, select five large and relatively random files from your hard drive (compressed log files are a good start) and put them under your /usr/share/ssl directory. These will act as your random seed enhancers. We refer to them as random1: random2:...: random5 below.

- To select five random files and put them under /usr/share/ssl, use the commands:

```
[root@deep /]# cp /var/log/boot.log /usr/share/ssl/random1
[root@deep /]# cp /var/log/cron /usr/share/ssl/random2
[root@deep /]# cp /var/log/dmesg /usr/share/ssl/random3
[root@deep /]# cp /var/log/messages /usr/share/ssl/random4
[root@deep /]# cp /var/log/secure /usr/share/ssl/random5
```

Step 3
Third, create the RSA private key **not protected with a pass-phrase** for the Samba server. The command below will generate 1024 bit RSA Private Key and stores it in the file smb.key.

- To generate the Key, use the following commands:

```
[root@deep /]# cd /usr/share/ssl/
[root@deep ssl]# openssl genrsa -rand
random1:random2:random3:random4:random5 -out smb.key 1024
123600 semi-random bytes loaded
Generating RSA private key, 1024 bit long modulus
.....................+++++
.....+++++
e is 65537 (0x10001)
Enter PEM pass phrase:
Verifying password - Enter PEM pass phrase:
```

WARNING: Please backup your **smb.key** file. A good choice is to backup this information onto a diskette or other removable media.

Step 4
Finally, generate a **C**ertificate **S**igning **R**equest (CSR) with the server RSA private key. The command below will prompt you for the X.509 attributes of your certificate. Remember to give the name smb.domain.com when prompted for '**Common Name**'. Do not enter your personal name here. We are requesting a certificate for a File Sharing Server, so the **Common Name** has to match the FQDN of your system.

- To generate the CSR, use the following command:

```
[root@deep ssl]# openssl req -new -key smb.key -out smb.csr
Using configuration from /usr/share/ssl/openssl.cnf
Enter PEM pass phrase:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
```

```
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [CA]:
State or Province Name (full name) [Quebec]:
Locality Name (eg, city) [Montreal]:
Organization Name (eg, company) [OpenNA, Inc.]:
Organizational Unit Name (eg, section) [File Sharing Server]:
Common Name (eg, YOUR name) [smb.openna.com]:
Email Address [noc@openna.com]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:.
An optional company name []:.
```

WARNING: Make sure you enter the FQDN (**F**ully **Q**ualified **D**omain **N**ame) of the server when OpenSSL prompts you for the "**Common Name**" (i.e. when you generate a CSR for a File Sharing Server which will be later accessed via smb.domain.com, enter smb.domain.com here).

After generation of your **C**ertificate **S**igning **R**equest (CSR), you could send this certificate to a commercial **C**ertifying **A**uthority (CA) like Thawte or Verisign for signing. You usually have to post the CSR into a web form, pay for the signing, await the signed certificate and store it into a smb.crt file. The result is then a real certificate, which can be used for Samba.

Step 5
You are not obligated to send your **C**ertificate **S**igning **R**equest (CSR) to a commercial **C**ertifying **A**uthority (CA) for signing. In some cases and with Samba you can become your own **C**ertifying **A**uthority (CA) and sign your certificate by yourself. In the step below, I assume that your CA keys pair, which is required for signing certificate by yourself, already exists on the server, if this is not the case, please refer to the chapter related to OpenSSL in this book for more information about how to create your CA keys pair and become your own **C**ertifying **A**uthority (CA).

- To sign server CSR's in order to create real SSL Certificates, use the following command:

```
[root@deep ssl]# /usr/share/ssl/misc/sign smb.csr
CA signing: smb.csr -> smb.crt:
Using configuration from ca.config
Enter PEM pass phrase:
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName           :PRINTABLE:'CA'
stateOrProvinceName   :PRINTABLE:'Quebec'
localityName          :PRINTABLE:'Montreal'
organizationName      :PRINTABLE:'OpenNA, Inc.'
organizationalUnitName:PRINTABLE:'File Sharing server'
commonName            :PRINTABLE:'smb.openna.com'
emailAddress          :IA5STRING:'noc@openna.com'
Certificate is to be certified until Jun 26 04:45:47 2003 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

```
CA verifying: smb.crt <-> CA cert
smb.crt: OK
```

This signs the CSR and results in a **smb.crt** file.

Step 6
Now, we must place the certificates files (smb.key and smb.crt) to the appropriate directories and change their default permission modes to be (0400/-r--------), owned by the super-user 'root' for Samba to be able to find and use them when it will start its daemon.

- To place the certificates into the appropriate directory, use the following commands:

```
[root@deep ssl]# mv smb.key private/
[root@deep ssl]# mv smb.crt certs/
[root@deep ssl]# chmod 400 private/smb.key
[root@deep ssl]# chmod 400 certs/smb.crt
[root@deep ssl]# chown 0.0 private/smb.key
[root@deep ssl]# chown 0.0 certs/smb.crt
[root@deep ssl]# rm -f smb.csr
```

First we move the smb.key file to the private directory and the smb.crt file to the certs directory. After that we change the permission mode and ownership of both certificates to be only readable and owned by the super-user 'root' for security reason. Finally we remove the smb.csr file from our system since it is no longer needed.

Step 7
To allow SSL-enabled connections with Samba, we must specify some new options into the smb.conf file. Text in bold is the parts of the lines that must be customized and adjusted to satisfy your needs.

- Edit the **smb.conf** file (vi /etc/samba/smb.conf), and add the following lines:

```
[global]

        workgroup = OPENNA
        server string = OpenNA Samba Server
        encrypt passwords = True
        security = user
        smb passwd file = /etc/samba/smbpasswd
        log file = /var/log/samba/log.%m
        max log size = 0
        socket options = IPTOS_LOWDELAY TCP_NODELAY
        deadtime = 15
        getwd cache = Yes
        lpq cache time = 45
        domain master = Yes
        local master = Yes
        preferred master = Yes
        os level = 65
        dns proxy = Yes
        wins support = Yes
        name resolve order = wins lmhosts host bcast
        bind interfaces only = True
        interfaces = eth0 192.168.1.1/24 127.0.0.1
        hosts deny = ALL
        hosts allow = 192.168.1. 207.35.78. 127.0.0.1
        debug level = 1
        create mask = 0644
```

```
        directory mask = 0755
        oplocks = True
        level2 oplocks = True
        read raw = No
        write cache size = 262144
        ssl = Yes
        ssl CA certFile = /usr/share/ssl/certs/ca.crt
        ssl server cert = /usr/share/ssl/certs/smb.crt
        ssl server key = /usr/share/ssl/private/smb.key

[homes]
        comment = Home Directories
        browseable = No
        read only = Yes
        invalid users = root bin daemon sync nobody sys tty disk mem kmem

[printers]
        comment = Remote Printers
        path = /var/spool/samba
        browseable = No
        printable = Yes
        invalid users = root bin daemon sync nobody sys tty disk mem kmem

[tmp]
        comment = Temporary File Space
        path = /tmp
        read only = No
        valid users = smbadmin
        invalid users = root bin daemon sync nobody sys tty disk mem kmem
```

The "ssl" variable enables the entire SSL mode on the Samba server. The second variable "ssl CA certFile" defines where to look up and find the Certification Authorities (CA). The "ssl server cert" will specify where the file containing the server's certificate is located. The "ssl server key" will specify where the file containing the server's private key is located.

Step 8
The Samba SSL-enabled connections run by default on port 139 with smbd daemon. To allow external traffic through this port (139), we must add a new rule into our firewall script file for the File Sharing Server to accept external connections on the system. Please note that this is only required if you want to share your files through the Internet. For LAN this is not required at all.

Step 9
Finally, we must restart our Samba File Sharing Server for the changes to take effect.

- To restart Samba use the following command:

```
[root@deep /]# /etc/init.d/smb restart
Shutting down Samba SMB:       [OK]
Shutting down Samba NMB:       [OK]
Starting Samba SMB:            [OK]
Starting Samba NMB:            [OK]
```

Step 10
Now that Samba is started, it is time to verify if everytinhg run as espected. A good way to test whether Samba is working properly is to use the smbclient program.

- On the Samba server, enter the following command, substituting the appropriate share and user for a connection:

```
[root@deep /]# smbclient //localhost/tmp -U smbadmin -I 192.168.1.1
SSL: Certificate OK:
/C=CA/ST=Quebec/L=Montreal/O=OpenNA.com/OU=OpenNA.com File Sharing
Server/CN=smb.openna.com/Email=noc@openna.com
SSL: Certificate OK:
/C=CA/ST=Quebec/L=Montreal/O=OpenNA.com/OU=OpenNA.com File Sharing
Server/CN=smb.openna.com/Email=noc@openna.com
SSL: negotiated cipher: DES-CBC3-SHA
Password:
Domain=[OPENNA] OS=[Unix] Server=[Samba 2.2.4]
smb: \> exit
```

If you see several debugging statements followed by a line indicating the negotiated cipher, such as: "SSL: negotiated cipher: DES-CBC3-SHA", congratulations, your Samba File Sharing Server is working with SSL support enable.

Securing Samba

This section deals especially with actions we can make to improve and tighten security under Samba. The interesting points here are that we refer to the features available within the base installed program and not to any additional software.

Create the encrypted Samba password file for your client connections:

The /etc/samba/smbpasswd file is where the Samba encrypted passwords are stored. It contains the username; Unix UID and SMB hashed passwords of the allowed users to your Samba server, as well as account flag information and the time the password was last changed.

It's important to create this password file and include all allowed users to it before your client machines try to connect to your File Sharing Server. Without this step, no one will be able to connect to your Samba server.

Step 1
To create new Samba users accounts on the system, you must first have a valid Linux account for them, therefore it is important before generating the "smbpasswd" file of Samba which will handle all Samba users allowed to connect to the system, to create in /etc/passwd file all users you want to be able to connect to your Samba server.

- Use the following command to create new users in the /etc/passwd file. This step must be done on each additional user that you allow to access the File Sharing Server.

```
[root@deep /]# useradd -s /bin/false smbadmin
[root@deep /]# passwd smbadmin
Changing password for user smbadmin
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully
```

The useradd command will add the new Samba user called smbadmin to the File Sharing Server. The '-s' option specifies the name of the user's login shell in our case we choose /bin/false. Finally, the passwd command will set the password for this user 'smbadmin'.

Here it is important to make a special attention to the above command that I use to generate the Samba user account. If you remark, this user doesn't have a shell account on the system, he just have a valid username and password to log in and nothing else.

Step 2
Once we have added all Samba clients in our /etc/passwd file on the Linux server, we can now generate the "smbpasswd" file from the /etc/passwd file.

- To generate "smbpasswd" file from /etc/passwd file, use the following command:
 [root@deep /]# **cat /etc/passwd | mksmbpasswd.sh > /etc/samba/smbpasswd**

Step 3
Finally, the last step will be to create the same Samba user account in our new generated /etc/samba/smbpasswd file before we can use it.

- To create the same Samba user account, use the following command:
```
[root@deep /]# smbpasswd -a smbadmin
INFO: Debug class all level = 1   (pid 3123 from pid 3123)
New SMB password:
Retype new SMB password:
Password changed for user smbadmin.
```

Step 4
Don't forget to change the default permission mode of the new "smbpasswd" file to be readable and writable only by the super-user "root", and nothing for group and other (0600/-rw-------). This is a security measure.

```
[root@deep /]# chmod 600 /etc/samba/smbpasswd
[root@deep /]# testparm (this will verify the smb.conf file for possible error).
```

NOTE: See the file called "ENCRYPTION.txt" in samba/doc/texts/ for more information.

Use Anti-Virus scanner program:

We already know that Linux cannot be infected by virus but on a File Sharing Server, transferred files from Windows systems could be infected. To avoid the risk that some innocent users transfer some virus from Windows to Linux and share it with other users, we can use Anti-Virus scanner software like Sophos or F-Prot on the Linux File Sharing Server.

In this way, if some of your user transfers virus on the Samba server, the virus scanner program will detect and remove it before other possible users may transfer the infected files on their Windows system.

To archive this result, you only need to install Sophos or F-Prot on the Samba server and create a cron job which will run daily to scan all shared directories for possible infected files coming from Windows systems on the share server.

Immunize important configuration files:

The immutable bit can be used to prevent accidentally deleting or overwriting a file that must be protected. It also prevents someone from creating a symbolic link to this file. Once your "smb.conf" and "lmhosts" files have been configured, it's a good idea to immunize them with a command like:

```
[root@deep /]# chattr +i /etc/samba/smb.conf
[root@deep /]# chattr +i /etc/samba/lmhosts
```

Optimizing Samba

This section deals especially with actions we can make to improve and tighten performance of the Samba server. Take a note that we refer to the features available within the base installed program.

Get some fast SCSI hard disk:

Once again, one of the most important parts of optimizing a File Sharing Server as well as for the majority of all type of servers is the speed of your hard disk, the fastest it'll be, and the fastest your File Sharing Server will run. Considering a SCSI disk with low seek times like 4.2ms can make all the difference, much better performance can also be made with RAID technology.

Setting a "wide links" Samba parameter in configuration file:

It is a big mistake to set the "wide links" Samba parameter to "No" in the Samba configuration file /etc/samba/smb.conf. This option if set to "No", instructs Samba to not follow symbolic links outside of an area designated as being exported as a share point.

In order to determine if a link points is outside the shared area, Samba has to follow the link and then do a directory path lookup to determine where on the file system the link ended up. This ends up adding a total of six extra system calls per filename lookup, and Samba looks up filenames a lot.

A test done was published that showed that setting this parameter would cause a 25- to 30-percent slowdown in Samba performance. Therefore setting this parameter to "No" can have a negative effect on your server performance due to the extra system calls that Samba will have to do in order to perform the link checks.

Tuning the buffer cache:

The modification of the file system cache-tuning parameters can significantly improve Linux file-serving performance--up to a factor of two. Linux will attempt to use memory not being used for any other purpose for file system caching. A special daemon, called "bdflush", will periodically flush "dirty" buffers (buffers that contain modified file system data or metadata) to the disk.

The secret to good performance is to keep as much of the data in memory for as long as is possible. Writing to the disk is the slowest part of any file system. If you know that the file system will be heavily used, then you can tune this process for Linux Samba.

As with many kernel tunable options, this modification can be done on the fly by writing to special files in the /proc file system. The trick is you have to tell Linux you want it to do that. You do so by executing the following command.

The default setup for the "bdflush" parameters under Red Hat Linux is:
"30 64 64 256 500 3000 60 0 0"

Step 1
To change the values of **bdflush**, type the following command on your terminal:

- Edit the **sysctl.conf** file (vi /etc/sysctl.conf) and add the following line:

```
# Improve file system performance for Samba
vm.bdflush = 80 500 64 64 15 6000 6000 0 0
```

Step 2
You must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/init.d/network restart
Setting network parameters             [OK]
Bringing up interface lo               [OK]
Bringing up interface eth0             [OK]
Bringing up interface eth1             [OK]
```

The above modifications in the /proc file system tells "bdflush" not to worry about writing out dirty blocks to the disk until the file system buffer cache is 80 percent full (80). The other values tune such things as the number of buffers to write out in one disk operation (500), how long to allow dirty buffers to age in the kernel (60*HZ), etc.

NOTE: There is another way to update the entry without restarting the network by using the following command into your terminal screen:

```
[root@deep /]# sysctl -w vm.bdflush="80 500 64 64 15 6000 6000 0 0"
```

Tuning the buffermem:

Another helpful tuning hint is to tell Linux the following: Use a minimum of 60 percent of memory for the buffer cache; only prune when the percentage of memory used for the buffer cache gets over 10 percent (this parameter is now unused); and allow the buffer cache to grow to 60 percent of all memory (this parameter is also unused now).

The default setup for the buffermem parameters under Red Hat Linux is:
"2 10 60"

Step 1
To change the values of **buffermem**, type the following command on your terminal:

- Edit the **sysctl.conf** file (vi /etc/sysctl.conf) and add the following line:

```
# Improve virtual memory performance for Samba
vm.buffermem = 60 10 60
```

Step 2
You must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all networks devices manually on your system, use the following command:
```
[root@deep /]# /etc/init.d/network restart
Setting network parameters             [OK]
Bringing up interface lo               [OK]
Bringing up interface eth0             [OK]
Bringing up interface eth1             [OK]
```

Recall that the last two parameters (**10** and **60**) are unused by the system so we don't need to change the default ones.

NOTE: There is another way to update the entry without restarting the network by using the following command into your terminal screen:
```
[root@deep /]# sysctl -w vm.buffermem="60 10 60"
```

Samba Administrative Tools

The commands listed below are some that we use often, but many more exist. Check the manual pages and documentation of Samba for more information.

smbstatus

The smbstatus utility is a very simple program to list the current Samba connections.

- To report current Samba connections, use the following command:
```
[root@deep /]# smbstatus

INFO: Debug class all level = 1   (pid 7402 from pid 7402)
Samba version 2.2.4

Service      uid      gid      pid    machine
----------------------------------------------
IPC$         smbadmin smbadmin 2688   station1 (192.168.1.30) Wed Jun 12
13:22:32 2002
No locked files
```

Samba Users Tools

The commands listed below are some that we use often, but many more exist. Check the manual pages and documentation that comes with Samba for more information.

smbclient

The smbclient program utility for Samba works much like the interface of the FTP program. This small program allow you to get files from the server to the local machine, put files from the local machine to the server, retrieve directory information from the server, and so on.

- To connect to a Windows machine with smbclient utility, use the following command:

```
[root@deep /]# smbclient //station1/Tmp -U smbadmin –I 192.168.1.1
Password:
Domain=[OPENNA] OS=[Windows NT 5.0] Server=[NT LAN Manager 5.0]
smb: \> ls
  .                   D        0   Tue Mar 14 15:31:50 2001
  ..                  D        0   Tue Mar 14 15:31:50 2001
  PostgreSQL          D        0   Tue Mar 14 15:32:22 2001
  Squid               D        0   Tue Mar 14 15:32:28 2001
  Imap                D        0   Tue Mar 14 15:32:38 2001
  E_comm              D        0   Tue Mar 14 15:32:42 2001
  StackGuard.pdf      A    61440   Tue Dec 21 20:41:34 2001

                65510 blocks of size 32768. 5295 blocks available
smb: \>exit
```

Where "//station1" is the name of the server you want to connect to. "/Tmp" is the directory on this server you want to connect to, and "smbadmin" is your username on this machine. The "-I" option indicates to use the specified network interface for the connection.

Further documentation

For more details about Samba, there are several manual pages that you could read:

```
$ man Samba (7)          - A Windows SMB/CIFS fileserver for UNIX.
$ man smb.conf (5)       - The configuration file for the Samba suite.
$ man smbclient (1)      - An ftp-like client to access SMB/CIFS resources on servers.
$ man smbd (8)           - Server to provide SMB/CIFS services to clients.
$ man smbmnt (8)         - Mount smb file system.
$ man smbmount (8)       - Mount smb file system.
$ man smbpasswd (5)      - The Samba encrypted password file.
$ man smbpasswd (8)      - Change a users SMB password.
$ man smbrun (1)         - Interface program between smbd and external programs.
$ man smbsh (1)          - Allows access to Windows NT filesystem using UNIX commands.
$ man smbstatus (1)      - Report on current Samba connections.
$ man smbtar (1)         - Shell script for backing up SMB shares directly to UNIX tape drives.
$ man smbumount (8)      - Umount for normal users.
$ man testparm (1)       - Check an smb.conf configuration file for internal correctness.
$ man testprns (1)       - Check printer name for validity with smbd.
```



Part XVI

Backup

CHAPTER 47

Tar & Dump

IN THIS CHAPTER

1. Recommended `RPM` packages to be installed for a Backup Server
2. The tar backup program
3. Making backups with `tar`
4. Automating tasks of backups made with `tar`
5. Restoring files with `tar`
6. The `dump` backup program
7. Making backups with `dump`
8. Restoring files with `dump`
9. Backing up and restoring over the network

Linux Tar & Dump

Abstract
A secure and reliable server is closely related to performing regular backups. Failures will probably occur sometimes. They may be caused by attacks, hardware failure, human error, power outages, etc. The safest method of doing backups is to record them in a location separate from your Linux system like over a network, from tape, removable drive, writable CD-ROM, etc.

Many methods of performing backups with Linux exist, such as "dump", "tar", "cpio", as well as "dd" commands that are each available by default on your Linux system. Also available are text-based utilities program, such as "Amanda", which is designed to add a friendlier user interface to the backup and restore procedures. Finally, commercial backup utilities are also available, such as "BRU".

The procedures for performing a backup and restore will differ depending on your choice of a backup solution. For this reason we will discuss methods for performing backups with the traditional UNIX tools: "tar", and "dump" which is a command-line backup tool.

What to backup
The idea of making a backup is to back up as much as possible on your system, but some exceptions exist as shown below. It is not logical to include these in your backup since you will lose time and space in your media for nothing.

The major exceptions to not include in your backup are:

- ✓ The /proc file system: since it only contains data that the kernel generates automatically, it is never a good idea to back it up.
- ✓ The /mnt file system, because it is where you mount your removable media like CD-ROM, floppy disk and other.
- ✓ The backup directory or media where you have placed your backup files, such as a tape, CD-ROM, NFS mounted file system, remote/local directory or other kind of media.
- ✓ Software that can be easily reinstalled, though they may have configuration files that are important to back up, lest you do all the work to configure them all over again. I will recommend putting them (the configuration files for software) on the floppy disk.

The tar backup program
The tar backup program is an archiving program designed to store and extract files from an archive file known as a tarfile. A tarfile may be made on a tape drive; however, it is also common to write a tarfile to a normal file.

A simple backup scheme:

When you decide to make a backup of files on your system you must choose a backup scheme before the beginning of your backup procedure. A lot of strategic backup schemes exist, and depend on the backup policies you want to use. In the following, I will show you one backup scheme that you may use which takes advantage of the `tar` program's possibilities. This scheme is to first back up everything once, then back up everything that has been modified since the previous backup. The first backup is called a full backup; the subsequent ones are incremental backups.

Making backups with `tar`

With six tapes you can make backups every day; the procedure is to use tape 1 for the first full backup (Friday 1), and tapes 2 to 5 for the incremental backups (Monday through Thursday). Then, you make a new full backup on tape 6 (second Friday), and start doing incremental ones with tapes 2 to 5 again. It's important to keep tape 1 at its state until you've got a new full backup with tape 6. In the following example below, we assume that we write the backup to a `SCSI` tape drive named "`/dev/st0`", and we backup the home directory "`/home`" of our system.

First of all, we move to the file system "**/**" partition. When creating an archive file, `tar` will strip leading "`/`" (slash) characters from file path names. This means that restored files may not end up in the same locations they were backed up from. Therefore, to solve the problem, the solution is to change to the "**/**" root directory before making all backups and restorations.

- To move to the "`/`" root directory, use the command:
 `[root@deep]#` **`cd /`**

It is important to always start with a full backup (say, on a Friday), for example:

- Friday 1, (use tape 1 for the first full backup).
  ```
  [root@deep /]# cd /
  [root@deep /]# tar cpf /dev/st0 --label="full-backup created on \
  `date '+%d-%B-%Y'`." --directory / home
  ```

- Monday, (use tapes 2 for the incremental backups).
  ```
  [root@deep /]# cd /
  [root@deep /]# tar cpNf /dev/st0 --label="full-backup created on \
  `date '+%d-%B-%Y'`." --directory / home
  ```

- Tuesday, (use tapes 3 for the incremental backups).
  ```
  [root@deep /]# cd /
  [root@deep /]# tar cpNf /dev/st0 --label="full-backup created on \
  `date '+%d-%B-%Y'`." --directory / home
  ```

- Wednesday, (use tapes 4 for the incremental backups).
  ```
  [root@deep /]# cd /
  [root@deep /]# tar cpNf /dev/st0 --label="full-backup created on \
  `date '+%d-%B-%Y'`." --directory / home
  ```

- Thursday, (use tapes 5 for the incremental backups).
  ```
  [root@deep /]# cd /
  [root@deep /]# tar cpNf /dev/st0 --label="full-backup created on \
  `date '+%d-%B-%Y'`." --directory / home
  ```

- Friday 2, (use tape 6 for the new full backups).

```
[root@deep /]# cd /
[root@deep /]# tar cpf /dev/st0 --label="full-backup created on \
`date '+%d-%B-%Y'`." --directory / home
```

- Now, start doing incremental ones with tapes 2 to 5 again and so on.

The "c" option specifies that an archive file is beginning to be created.
The "p" option preserves permissions; file protection information will be "remembered".
The "N" option does an incremental backup and only stores files newer than DATE.
The "f" option states that the very next argument will be the name of the archive file or device being written.

Notice how a filename, which contains the current date, is derived, simply by enclosing the "date" command between two back-quote characters. A common naming convention is to add a "tar" suffix for non-compressed archives, and a "tar.gz" suffix for compressed ones. Since we aren't able to specify a filename for the backup set, the "--label" option can be used to write some information about the backup set into the archive file itself. Finally, only the files contained in the /home are written to the tape.

Because the tape drive is a character device, it is not possible to specify an actual file name. Therefore the file name used as an argument to tar is simply the name of the device /dev/st0, the first tape device. The /dev/st0 device does not rewind after the backup set is written; Therefore, it is possible to write multiple sets on one tape. You may also refer to the device as /dev/st0, in which case the tape is automatically rewound after the backup set is written. When working with tapes you can use the following commands to rewind and eject your tape:

```
[root@deep /]# mt -f /dev/st0 rewind
[root@deep /]# mt -f /dev/st0 offline
```

WARNING: To reduce the space needed on a tar archive, the backups can be compressed with the "z" option of tar program. Unfortunately, using this option to compress backups can cause trouble. Due to the nature of how compression works, if a single bit in the compressed backup is wrong, all the rest of the compressed data will be lost. It's recommended to NOT using compression (the "z" option) to make backups with the tar command.

- If your backup doesn't fit on one tape, you'll have to use the --multi-volume (-M) option:

```
[root@deep /]# cd /
[root@deep /]# tar cMpf /dev/st0 /home
Prepare volume #2 for /dev/st0 and hit return:
```

- After you have made a backup, you should check that it is OK, using the --compare (-d) option as shown below:

```
[root@deep /]# cd /
[root@deep /]# tar dvf /dev/st0
```

- To perform a backup of your entire system, use the following command:

```
[root@deep /]# cd /
[root@deep /]# tar cpf /archive/full-backup-`date '+%d-%B-%Y'`.tar \
 --directory / --exclude=proc --exclude=mnt --exclude=archive \
 --exclude=cache --exclude=*/lost+found .
```

The "`--directory`" option informs `tar` to first switch to the following directory path (the "`/`" directory in this example) prior to starting the backup. The "`--exclude`" options informs `tar` not to bother backing up the specified directories or files. Finally, the "`.`" character at the end of the command tells `tar` that it should back up everything in the current directory.

WARNING: When backing up your file systems, do not include the `/proc` pseudo-file-system! The files in `/proc` are not actually files but are simply file-like links which describe and point to kernel data structures. Also, do not include the `/mnt`, `/archive`, and all `lost+found` directories.

Automating tasks of backups made with `tar`

It is always interesting to automate the tasks of a backup. Automation offers enormous opportunities for using your Linux server to achieve the goals you set. The following example below is our backup script, named "`backup.cron`".

This script is designed to run on any computer by changing only the four variables: `COMPUTER`, `DIRECTORIES`, `BACKUPDIR`, and `TIMEDIR`. We suggest that you set this script up and run it at the beginning of the month for the first time, and then run it for a month before making major changes. In our example below we do the backup to a directory on the local server (`BACKUPDIR`), but you could modify this script to do it to a tape on the local server or via an `NFS` mounted file system.

Step 1
Create the backup script **`backup.cron`** file (`touch /etc/cron.daily/backup.cron`) and add the following lines to this backup file:

```
#!/bin/sh
# Full and incremental backup script
# Updated 04 July 2002
# Based on a script by Daniel O'Callaghan <danny@freebsd.org>
# and modified by Gerhard Mourani <gmourani@openna.com>

# Change the 5 variables below to fit your computer/backup

COMPUTER=deep                   # Name of this computer
DIRECTORIES="/home"             # Directory to backup
BACKUPDIR=/backups              # Where to store the backups
TIMEDIR=/backups/last-full      # Where to store time of full backup
TAR=/bin/tar                    # Name and location of tar

# You should not have to change anything below here

PATH=/usr/local/bin:/usr/bin:/bin
DOW=`date +%a`                  # Day of the week e.g. Mon
DOM=`date +%d`                  # Date of the Month e.g. 27
DM=`date +%d%b`                 # Date and Month e.g. 27 Sep

# On the 1st of the month a permanet full backup is made
# Every Sunday a full backup is made - overwriting last Sundays backup
# The rest of the time an incremental backup is made. Each incremental
# backup overwrites last weeks incremental backup of the same name.
#
# if NEWER = "", then tar backs up all files in the directories
# otherwise it backs up files newer than the NEWER date. NEWER
# gets it date from the file written every Sunday.
```

```
# Monthly full backup
if [ $DOM = "01" ]; then
        NEWER=""
        $TAR $NEWER -cf $BACKUPDIR/$COMPUTER-$DM.tar $DIRECTORIES
fi

# Weekly full backup
if [ $DOW = "Sun" ]; then
        NEWER=""
        NOW=`date +%d-%b`

        # Update full backup date
        echo $NOW > $TIMEDIR/$COMPUTER-full-date
        $TAR $NEWER -cf $BACKUPDIR/$COMPUTER-$DOW.tar $DIRECTORIES

# Make incremental backup - overwrite last weeks
else

        # Get date of last full backup
        NEWER="--newer `cat $TIMEDIR/$COMPUTER-full-date`"
        $TAR $NEWER -cf $BACKUPDIR/$COMPUTER-$DOW.tar $DIRECTORIES
fi
```

Here is an abbreviated look of the backup directory after one week:

```
[root@deep /]# ls -l /backups/
total 22217
-rw-r--r--  1 root     root         10731288    Feb  7 11:24 deep-01Feb.tar
-rw-r--r--  1 root     root             6879    Feb  7 11:24 deep-Fri.tar
-rw-r--r--  1 root     root             2831    Feb  7 11:24 deep-Mon.tar
-rw-r--r--  1 root     root             7924    Feb  7 11:25 deep-Sat.tar
-rw-r--r--  1 root     root         11923013    Feb  7 11:24 deep-Sun.tar
-rw-r--r--  1 root     root             5643    Feb  7 11:25 deep-Thu.tar
-rw-r--r--  1 root     root             3152    Feb  7 11:25 deep-Tue.tar
-rw-r--r--  1 root     root             4567    Feb  7 11:25 deep-Wed.tar
drwxr-xr-x  2 root     root             1024    Feb  7 11:20 last-full
```

WARNING: The directory where to store the backups (`BACKUPDIR`), and the directory where to store time of full backup (`TIMEDIR`) must exist or be created before the use of the backup-script, or you will receive an error message.

Also I recommend you to set the permission mode of these directories to be (`0700/-rwx------`) owned by the user making the backup. It is important that normal user cannot access in our example the `/backups` directory.

Step 2
If you are not running this backup script from the beginning of the month (01-month-year), the incremental backups will need the time of the Sunday backup to be able to work properly. If you start in the middle of the week, you will need to create the time file in the `TIMEDIR`.

- To create the time file in the `TIMEDIR` directory, use the following command:
 `[root@deep /]#` **`date +%d%b > /backups/last-full/myserver-full-date`**

Where <`/backups/last-full`> is our variable `TIMEDIR` where we want to store the time of the full backup, and <`myserver-full-date`> is the name of our server (e.g., `deep`), and our time file consists of a single line with the present date (i.e. `15-Feb`).

Step 3
Make this script executable and change its default permissions to be writable only by the super-user "`root`" (`0700/-rwx------`).

```
[root@deep /]# chmod 700 /etc/cron.daily/backup.cron
```

NOTE: Because this script is in the `/etc/cron.daily` directory, it will be automatically run as a `cron` job at one o'clock in the morning every day.

Restoring files with `tar`

More important than performing regular backups is having them available when we need to recover important files! In this section, we will discuss methods for restoring files, which have been backed up with "`tar`" command.

The following command will restore all files from the "`full-backup-Day-Month-Year.tar`" archive, which is an example backup of our `/home` directory created from the example `tar` commands shown above.

- To restore a full backup of the `/home` directory, use the following commands:
 `[root@deep /]#` **`cd /`**
 `[root@deep /]#` **`tar xpf /dev/st0/full-backup-Day-Month-Year.tar`**

The above command extracts all files contained in the compressed archive, preserving original file ownership and permissions.

The "`x`" option stands for extract.
The "`p`" option preserves permissions; file protection information will be "remembered".
The "`f`" option states that the very next argument will be the name of the archive file or device.

If you do not need to restore all the files contained in the archive, you can specify one or more files that you wish to restore:

- To specify one or more files that you wish to restore, use the following commands:
```
[root@deep]# cd /
[root@deep]# tar xpf /dev/st0/full-backup-Day-Month-Year.tar \
 home/wahib/Personal/Contents.doc home/quota.user
```

The above command restores the `/home/wahib/Personal/Contents.doc` and `/home/quota.user` files from the archive.

- If you just want to see what files are in the backup volume, Use the `--list` (`-t`) option:
```
[root@deep /]# tar tf /dev/st0
```

WARNING: If you have files on your system set with the immutable bit, using the "`chattr`" command, these files will not be remembered with the immutable bit from your restored backup. You must reset it immutable with the command "`chattr +i`" after the backup is completed.

Testing the ability to recover from backups:

For many system administrators, recovering a file from a backup is an uncommon activity. This step assures that if you need to recover a file, the tools and processes will work. Performing this test periodically will help you to discover problems with the backup procedures so you can correct them before losing data.

Some backup restoration software does not accurately recover the correct file protection and file ownership controls. Check the attributes of restored files to ensure they are being set correctly. Periodically test to ensure that you can perform a full system recovery from your backups.

Further documentation

For more details, there is one manual page about `tar` that you could read:

`tar (1)` - The GNU version of the tar archiving utility

The dump backup program

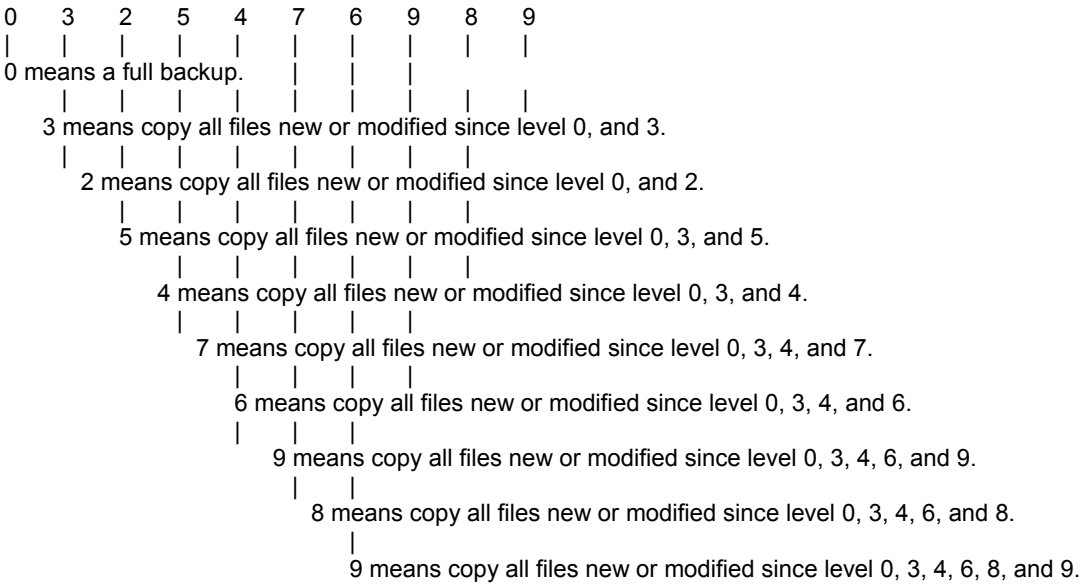
Dump is completely different from tar; it is a program for backing up and restoring file system. It backups up the entire file system - not the files. Dump does not care what file system is on the hard drive, or even if there are files in the file system. It examines files on an ext3 file system, determines which ones need to be backed up, and copies those files to a specified disk, tape, file or other storage medium. It dumps one file system at a time quickly and efficiently.

Unfortunately, it does not do individual directories, and so it eats up a great deal more storage space than tar. It is also written specifically for backups. The restore command performs the inverse function of dump; It can restore a full backup of a file system. Subsequent incremental backups can then be layered on top of the full backup. Single files and directory sub trees may also be restored from full or partial backups. You can use dump if you need a procedure for both backing up file systems and restoring file systems after backups.

The Dump levels:

Dump has several levels of backup procedures. The levels range from 0 to 9, where level number 0 means a full backup and guarantees the entire file system is copied. A level number above 0, incremental backup, tells dump to copy all files new or modified since the last dump of the same or lower level. To be more precise, at each incremental backup level you back up everything that has changed since the previous backup at the same or a previous level.

What are the advantages and the reasons to create and use several levels to make a backup? I try to explain it with the following schemas:



The advantages and reasons for doing this are that with multiple levels, the backup history can be extended more cheaply. A longer backup history is useful, since deleted or corrupted files are often not noticed for a long time. Even a version of a file that is not very up to date is better than no file at all. Also, backup levels are used to keep both the backup and restore times to a minimum (low).

The dump manual page suggests a good scheme to take the full advantage of backup levels: 3, 2, 5, 4, 7, 6, 9, 8, 9, etc as described by the table below. The most you have to backup is two day's worth of work. The number of tapes for a restore depends on how long you keep between full backups.

Tape	Level	Backup (days)	Restore tapes
1	0	n/a	1
2	3	1	1, 2
3	2	2	1, 3
4	5	1	1, 2, 4
5	4	2	1, 2, 5
6	7	1	1, 2, 5, 6
7	6	2	1, 2, 5, 7
8	9	1	1, 2, 5, 7, 8
9	8	2	1, 2, 5, 7, 9
10	9	1	1, 2, 5, 7, 9, 10

Making backups with dump

It's interesting to use the dump backup program if you want to take advantage of its several levels of backup procedures. Below, I show you a procedure to have a longer backup history, and to keep both the backup and restore times to a minimum.

In the following example, we assume that we write the backup to a tape drive named "/dev/st0" and we backup the /home directory of our system.

It is important to always start with a level 0 backup, for example:

- Friday 1, (use tape 1 for the first full backup).

```
[root@deep /]# dump -0u -f /dev/st0 /home
  DUMP: Date of this level 0 dump: Fri Mar 16 21:25:12 2001
  DUMP: Date of last level 0 dump: the epoch
  DUMP: Dumping /dev/sda6 (/home) to /dev/st0
  DUMP: mapping (Pass I) [regular files]
  DUMP: mapping (Pass II) [directories]
  DUMP: estimated 18582 tape blocks on 0.48 tape(s).
  DUMP: Volume 1 started at: Fri Mar 16 21:25:12 2001
  DUMP: dumping (Pass III) [directories]
  DUMP: dumping (Pass IV) [regular files]
  DUMP: DUMP: 18580 tape blocks on 1 volumes(s)
  DUMP: finished in 4 seconds, throughput 4645 KBytes/sec
  DUMP: Volume 1 completed at: Fri Mar 16 21:26:12 2001
  DUMP: Volume 1 took 0:00:04
  DUMP: Volume 1 transfer rate: 4645 KB/s
  DUMP: level 0 dump on Fri Fri Mar 16 21:25:12 2001
  DUMP: DUMP: Date of this level 0 dump: Fri Mar 16 21:25:12 2001
  DUMP: DUMP: Date this dump completed:  Fri Mar 16 21:25:18 2001
  DUMP: DUMP: Average transfer rate: 4645 KB/s
  DUMP: Closing /dev/st0
  DUMP: DUMP IS DONE
```

- Monday, (use tapes 2 for the incremental backups).
```
[root@deep /]# dump -3u -f /dev/st0 /home
```

- Tuesday, (use tapes 3 for the incremental backups).
```
[root@deep /]# dump -2u -f /dev/st0 /home
```

- Wednesday, (use tapes 4 for the incremental backups).
```
[root@deep /]# dump -5u -f /dev/st0 /home
```

- Thursday, (use tapes 5 for the incremental backups).
```
[root@deep /]# dump -4u -f /dev/st0 /home
```

- Friday 2, (use tape 6 for the incremental backups).
```
[root@deep /]# dump -7u -f /dev/st0 /home
```

- Monday, (use tapes 2 for the incremental backups).
```
[root@deep /]# dump -3u -f /dev/st0 /home
```

- Tuesday, (use tapes 3 for the incremental backups).
```
[root@deep /]# dump -2u -f /dev/st0 /home
```

- Wednesday, (use tapes 4 for the incremental backups).
```
[root@deep /]# dump -5u -f /dev/st0 /home
```

- Thursday, (use tapes 5 for the incremental backups).
```
[root@deep /]# dump -4u -f /dev/st0 /home
```

- Friday 3, (use tape 7 for the incremental backups).
```
[root@deep /]# dump -6u -f /dev/st0 /home
```

- Monday, (use tapes 2 for the incremental backups).
```
[root@deep /]# dump -3u -f /dev/st0 /home
```

- Tuesday, (use tapes 3 for the incremental backups).
```
[root@deep /]# dump -2u -f /dev/st0 /home
```

- Wednesday, (use tapes 4 for the incremental backups).
```
[root@deep /]# dump -5u -f /dev/st0 /home
```

- Thursday, (use tapes 5 for the incremental backups).
```
[root@deep /]# dump -4u -f /dev/st0 /home
```

- Friday 4, (use tape 8 for the incremental backups only if there have 5 Fridays in one month).
```
[root@deep /]# dump -9u -f /dev/st0 /home
```

- Monday, (use tapes 2 for the incremental backups only if there have 5 Fridays in one month).
```
[root@deep /]# dump -3u -f /dev/st0 /home
```

- Tuesday, (use tapes 3 for the incremental backups only if there have 5 Fridays in one month).
```
[root@deep /]# dump -2u -f /dev/st0 /home
```

- Wednesday, (use tapes 4 for the incremental backups only if there have 5 Fridays in one month).
```
[root@deep /]# dump -5u -f /dev/st0 /home
```

- Thursday, (use tapes 5 for the incremental backups only if there have 5 Fridays in one month).
```
[root@deep /]# dump -4u -f /dev/st0 /home
```

- Month, (use another tape for a new full backup when the month change).
```
[root@deep /]# dump -0u -f /dev/st0 /home
```

Where "`-0` to `-9`" is the backup level option you want to use, the "`u`" option means to update the file `/etc/dumpdates` after a successful `dump`, the "`-f`" option to write the backup to file; the file may be a special device file like `/dev/st0` (a tape drive), `/dev/rsd1c` (a disk drive), an ordinary file, or "`-`" (the standard output). Finally, you must specify what you want to backup. In our example, it is the `/home` directory.

You can see that we use the same tapes 2 to 5 for daily backups (Monday to Thursday = **4** tapes), tapes 6, 7, and 8 for weekly backups (other Fridays, 6 + 7 + 8 = **3** tapes; note that there can be five Fridays in one month) and tapes 1 and any subsequent new one for monthly backups (first Friday each month, **1** + any subsequent "11 months" = 12 tapes). In conclusion, if we use 8 tapes (**4** + **3** + **1** = **8**), we can have a full backup for one month and repeat the procedure with the 8 tapes to get our subsequent 11 months to come for a total of 1-year individual full backups.

The full backup should be done at set intervals, say once a month, and on a set of fresh tapes that are saved forever. With this kind of procedure, you will have 12 tapes for 12 months that handle histories and changes of your system for one year. Afterwards, you can copy the 12 tape backups onto a different computer designated to keep all yearly backups for a long time and be able to reuse them (12 tapes) to repeat the procedure for a new year. Thank you Gerhard!

Restoring files with `dump`

The `restore` command of the program performs the inverse function of `dump(8)`. It restores files or file systems from backups made with `dump`. A full backup of a file system may be restored, and subsequent incremental backups layered on top of it. Single files and directory sub-trees may be restored from full, or partial, backups. You have a number of possible commands and options to restore backed up data with the `dump` program. Below, we show you a procedure that uses the full potential of the `restore` program with the most options possible. It is also done in interactive mode.

In an interactive restoration of files from a `dump`, the `restore` program provides a shell like interface that allows the user to move around the directory tree selecting files to be extracted, after reading in the directory information from the `dump`. The following is what we will see if we try to restore our `/home` directory:

First of all, we must move to the partition file system where we want to restore our backup. This is required, since the interactive mode of the `restore` program will restore our backups from the current partition file system where we have executed the `restore` command.

- To move to the partition file system we want to restore (the `/home` directory in our case), use the following command:
```
[root@deep /]# cd /home
```

- To restore files from a dump in interactive mode, use the following command:
```
[root@deep /home]# restore -i -f /dev/st0
restore >
```

A prompt will appear in your terminal, to list the current, or specified, directory. Use the "**ls**" command as shown below:

```
restore > ls
.:
admin/       lost+found/  named/       quota.group  quota.user   wahib/

restore >
```

To change the current working directory to the specified one, use the "**cd**" commands (in our example, we change to *wahib* directory) as shown below:

```
restore > cd wahib
restore > ls
./wahib:
.Xdefaults      .bash_logout    .bashrc
.bash_history   .bash_profile   Personal/

restore >
```

To add the current directory or file to the list of files to be extracted, use the "**add**" command (If a directory is specified, then it and all its descendents are added to the extraction list) as shown below:

```
restore > add Personal/
restore >
```

Files that are on the extraction list are marked with a "*****" when they are listed by the "**ls**" command:

```
restore > ls
./wahib:
 .Xdefaults      .bash_logout    .bashrc
 .bash_history   .bash_profile  *Personal/
```

To delete the current directory or specified argument from the list of files to be extracted, use the "**delete**" command (If a directory is specified, then it and all its descendents are deleted from the extraction list) as shown below:

```
restore > cd Personal/
restore > ls
./wahib/Personal:
*Ad?le_Nakad.doc        *Overview.doc
*BIMCOR/                *Resume/
*My Webs/               *SAMS/
*Contents.doc           *Templates/
*Divers.doc             *bruno universite.doc
*Linux/                 *My Pictures/

restore > delete Resume/
restore > ls
./wahib/Personal:
*Ad?le_Nakad.doc        *Overview.doc
*BIMCOR/                Resume/
*My Webs/               *SAMS/
```

```
*Contents.doc           *Templates/
*Divers.doc             *bruno universite.doc
*Linux/                 *My Pictures/
```

NOTE: The most expedient way to extract most of the files from a directory is to add the directory to the extraction list and then delete those files that are not needed.

To extract all files in the extraction list from the dump, use the "**extract**" command (Restore will ask which volume the user wishes to mount. The fastest way to extract a few files is to start with the last volume and work towards the first volume) as shown below:

```
restore > extract
You have not read any tapes yet.
Unless you know which volume your file(s) are on you should start
with the last volume and work towards the first.
Specify next volume #: 1
set owner/mode for '.'? [yn] y
```

To exit from the interactive restore mode after you have finished extracting your directories or files, use the "**quit**" command as shown below.

```
/sbin/restore > quit
```

NOTE: Other methods of restoration exist with the dump program; consult the manual page of dump for more information.

Further documentation

For more details, there is some manual pages related to program dump that you could read:

```
$ man dump (8)          - ext2 file system backup
$ man restore (8)       - Restore files or file systems from backups made with dump
```

Backing up and restoring over the network

Backups allow you to restore the availability and integrity of information resources following security breaches and accidents. Without a backup, you may be unable to restore a computer's data after system failures and security breaches.

It is important to develop a plan that is broad enough to cover all the servers you plan to deploy. We must determine what categories of files will be backed up. For example, you may choose to back up only user data files (i.e. /home) because damaged system files should be reloaded from the original distribution media.

There are common technological approaches to file backups. For network servers, an authoritative version of the informational content of the server is created and maintained on a secure machine that is backed up. If the server is compromised and its content damaged, it can be reloaded from the secure system maintaining the authoritative version. This approach is typically used for public servers, such as web servers, because the content changes at more predictable intervals.

It is important to ensure that backups are performed in a secure manner and that the contents of the backups remain secure. We recommend that the plan specify that:

- ✓ The source data is encrypted before being transmitted to the storage medium.
- ✓ The data remains encrypted on the backup storage media.
- ✓ The storage media are kept in a physically secure facility that is protected from man-made and natural disasters.

Transfer your backup in a secure manner over the network:

In the previous sections, we have shown you how to make a backup onto both a tape and files from the same system where you execute the backup procedure, with utilities like `tar` and `dump`. These programs (`tar` and `dump`) are capable of making backups over the network as well.

To be able to backup over the network, usually you must ensure that the insecure `RPM` packages named "`rmt`" and "`rsh`" are installed on your system. The "`rmt`" utility provides remote access to tape devices for programs like `dump`, and `tar`. To complement this, the "`rsh`" package contains a set of programs, which allow users to run commands on remote machines, login to other machines and copy files between machines (`rsh`, `rlogin` and `rcp` are this set of programs).

Since "`rsh`" can be easily hacked, and "`rmt`" depends on "`rsh`" to be able to work, we have chosen to not install them in our setup installation (see chapter related to Linux installation in this book for more information on the subject) for security reasons. Therefore, we must find another way to make backups over the network in a secure manner.

`SSH` technology is the solution for our problem (see chapter related to `OpenSSH` in this book for more information on the subject) because it also has the ability to copy data across the network with its "`scp`" command, through encryption. The following is a method that permits us to use the potential of `SSH` software to transfer our backups made with `tar` or `dump` in a secure manner via the "`scp`" `SSH` utility.

Using the `scp` command of `SSH` to transfer backups over the network:

The `scp` command copies files between hosts on a network. It uses `SSH` for data transfer, and uses the same authentication, and provides the same security, as `SSH`. Unlike the "`rcp`" utility that comes with the `RPM` package "`rsh`", "`scp`" will transmit your data over the network encrypted. In our example below, we transfer a backup file made with the `tar` archive program; the procedure to transfer a backup file or tape made with `dump` program is exactly the same.

Step 1
Before going into the command line that will transfer our data encrypted through the network, it is important to recall that `scp` command like any other `SSH` command used for encrypted connection between servers will ask us by default to enter a pass-phrase. This is not useful when we want to automate backup using `SSH` for the transfer. Fortunately, it is possible to configure `SSH` to not ask for the pass-phrase before establishing the remote encrypted connection. We do it my creating a new `SSH` user without a pass-phrase. Of course I suppose that this user already exist in your Unix `/etc/passwd` file. If you don't understand what I mean, please refer to the chapter related to `OpenSSH` in this book for more information on the subject.

- To create a new SSH user without a pass-phrase, use the following commands:

```
[root@deep /]# su backadmin
[backadmin@deep /]$ ssh-keygen -d
Generating DSA parameter and key.
Enter file in which to save the key (/home/backadmin/.ssh/id_dsa):
Created directory '/home/backadmin/.ssh'.
Enter passphrase (empty for no passphrase): < Here you press enter
Enter same passphrase again: < Here you press enter again
Your identification has been saved in /home/backadmin/.ssh/id_dsa.
Your public key has been saved in /home/backadmin/.ssh/id_dsa.pub.
The key fingerprint is:
1f:af:aa:22:0a:21:85:3c:07:7a:5c:ae:c2:d3:56:64 backadmin@deep
```

As we can see here, our new SSH user is named "backadmin" and already exist into the /etc/passwd file of the Linux system. We sudo to this user and generate a new keys pair for him. The most important part here, is when the program ask us to enter a pass-phrase, therefore we just press [Enter] to inform it that we don't want a pass-phrase for this new SSH user.

Step 2

Once the keys pair of our new SSH user have been generated, we must copy its local public key **id_dsa.pub** from its /home/backadmin/.ssh directory remotely into the server from where we want to make the secure connection for transferring the backup files under the name, say, "authorized_keys". One way to copy the file is to use the ftp command or you might need to send the public key in electronic mail to the administrator of the system. Just include the contents of the ~/.ssh/id_dsa.pub file in the message.

> **WARNING:** Don't forget that the same username in our case "backadmin" must exist on the other server side. This is required only to create the ~/.ssh directory required to place the public key.

Step 3

Now, we must edit the /etc/ssh/ssh_config file on the REMOTE host from where we have sent our **id_dsa.pub** key which has become **authorized_keys** and add some additional lines to its ssh_config file to allow our new SSH user to connect and transfer backup files without a pass-phrase to the server. The text in bold are the parts of the configuration file that must be customized and adjusted to satisfy your needs

- Edit the **ssh_config** file (vi /etc/ssh/ssh_config) on REMOTE server and add the following lines:

```
# Site-wide defaults for various options

 Host *
   ForwardAgent no
   ForwardX11 no
   RhostsAuthentication no
   RhostsRSAAuthentication no
   RSAAuthentication yes
   PasswordAuthentication no
   FallBackToRsh no
   UseRsh no
   BatchMode no
   CheckHostIP yes
```

```
  StrictHostKeyChecking yes
  IdentityFile ~/.ssh/identity
  IdentityFile ~/.ssh/id_dsa
  IdentityFile ~/.ssh/id_rsa1
  IdentityFile ~/.ssh/id_rsa2
  Port 22
  Protocol 2,1
  Cipher blowfish
  EscapeChar ~

Host 207.35.78.13
  ForwardAgent no
  ForwardX11 no
  RhostsAuthentication no
  RhostsRSAAuthentication no
  RSAAuthentication no
  PasswordAuthentication no
  FallBackToRsh no
  UseRsh no
  BatchMode yes
  CheckHostIP no
  StrictHostKeyChecking yes
  IdentityFile ~/.ssh/identity
  IdentityFile ~/.ssh/id_dsa
  IdentityFile ~/.ssh/id_rsa1
  IdentityFile ~/.ssh/id_rsa2
  Port 22
  Protocol 2,1
  Cipher blowfish
  EscapeChar ~
```

From what we can see, is that we have added a copy of the first configuration but have changed two important options. The "**`BatchMode yes`**" option allow to connect without a pass-phrase and the "**`Host 207.35.78.13`**" option specifies that only connection coming from `IP` address `207.35.78.13` (this is the one that we will use with the `scp` command to transfer the backup files) is allowed to use this configuration where users can connect without a pass-phrase. The other settings are the same as for the original one. Finally we keep the original setting for regular connection to the server where pass-phrase is required.

Step 4
After that, we edit the `/etc/ssh/sshd_config` file on REMOTE again, and add to the "`AllowUsers`" option, our new `SSH` user to allow him to connect to the REMOTE server.

- Edit the **`sshd_config`** file (`vi /etc/ssh/sshd_config`) on REMOTE server and change for example the following lines:

  ```
  AllowUsers gmourani
  ```

 To read:

  ```
  AllowUsers gmourani backadmin
  ```

Here we add our user named "`backadmin`" to the list of allowed user on the REMOTE host.

NOTE: Step 1 to step 4 must be made on each servers from where you want to establish an encrypted remote connection without a pass-phrase to transfer backup over the network.

Step 5
Finally, everything is supposed to be fine now and we are ready to transfer backup over the network in a secure way.

- To use `scp` to copy a backup tape or file to a remote secure system, use the command:
 `[backadmin@deep /]#` **`scp <localdir/to/filelocation>\ <user@host:/dir/for/file>`**

Where <`localdir/to/filelocation`> is the directory where your backup file resides on your LOCAL server, and <`user@host:/dir/for/file`> represents, in order, the username (user) of the person on the REMOTE site that will hold the backup file, the hostname (host) of the remote host where you want to send the backup file, and the remote directory of this host where you want to place the transferred backup file.

A real example will look like this:

```
[backadmin@deep /]# scp -Cp /backups/deep-01Feb.tar \
backadmin@backupserver:/archive/deep/deep-01Feb.tar
deep-01Feb.tgz      |      10479 KB | 154.1 kB/s | ETA: 00:00:00 | 100%
```

NOTE: The "`C`" option enables compression for fast data transfer over the encrypted session, the "`p`" option indicates that the modification and access times as well as modes of the source file should be preserved on the copy. This is usually desirable. It is important to note that the <`dir/for/file`> directory on the remote host (`/archive/deep` in our example) must be owned by the "username" you specify in your `scp` command ("`admin`" is this username in our example) or you may receive error message like: `scp: /archive/deep/deep-01Feb.tar: Permission denied`.

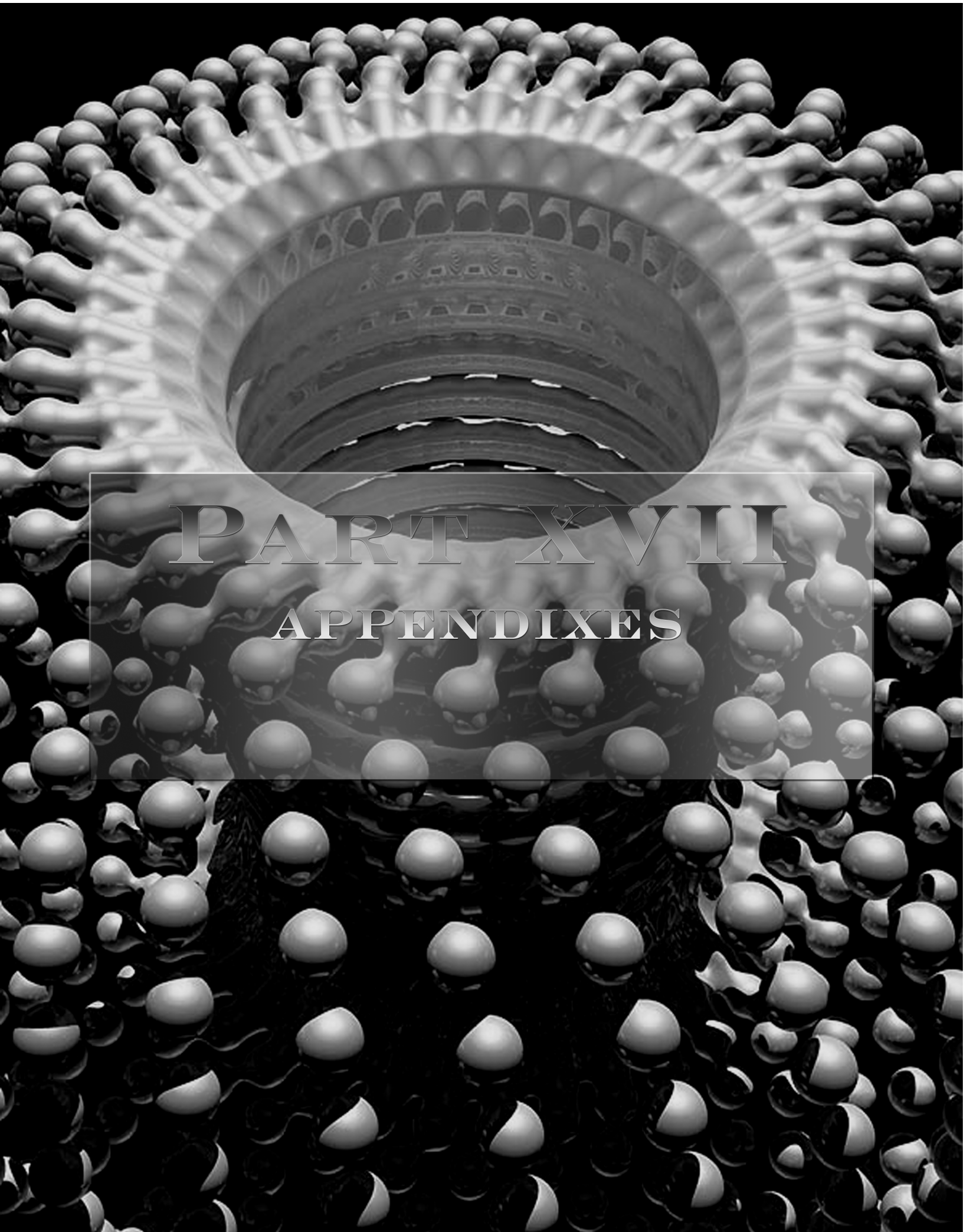
- To use `scp` to copy a remote tape or file to the local system, use the command:
 `[backadmin@deep /]#` **`scp <user@host:/dir/for/file>\ <localdir/to/filelocation>`**

Where <`user@host:/dir/for/file`> represents, in order, the username (user) of the person on the REMOTE site that holds the backup file, the hostname (host) of the REMOTE host where you want to get the backup file, and the REMOTE directory of this host where the backup file is kept, and <`localdir/to/filelocation`> is the LOCAL directory on your system where your want to place the backup file that you get from the REMOTE host.

A real example would look like this:

```
[backadmin@deep /]# scp -Cp admin@backupserver:/archive/deep/deep-
01Feb.tar /backups
admin@backupserver's password:
deep-01Feb.tgz      |      10479 KB | 154.1 kB/s | ETA: 00:00:00 | 100%
```

NOTE: It is important to note that the <`localdir/to/filelocation`> directory on the LOCAL host ("`/backups`" in our example) must be owned by the "username" you specify in your `scp` command ("`admin`" is this username in our example) or you may receive an error message like: `scp: /backups/deep-01Feb.tar: Permission denied`.



Part XVII

Appendixes

APPENDIX A
Tweaks, Tips and Administration tasks

Tweaks, Tips and Administration tasks

Some of the tips in this section are specific to Linux systems. Most are applicable to UNIX system in general. I make this section available since I think that it can be useful in daily administrative tasks from most of us.

1.0 The `du` utility command:

You can use the **`du`** utility to estimate file space usage. For example, to determine in megabyte the sizes of the `/var/log` and `/home` directories trees, type the following command:

```
[root@deep /]# du -sh /var/log /home
3.5M        /var/log
350M        /home
```

Keep in mind that the above command will report the actual size of your data. Now that you know for example that `/home` is using 350M you can move into it and `du -sh *` to locate where the largest files are.

```
[root@deep /]# cd /home/
[root@deep /home]# du -sh *
343M        admin
11k         ftp
6.8M        httpd
12k         lost+found
6.0k        named
6.0k        smbclient
6.0k        test
8.0k        www
```

NOTE: You can add this command to your crontab so that every day you get emailed the desired disk space list, and you'll be able to monitor it without logging in constantly.

1.1 Find the route that the packets sent from your machine to a remote host:

If you want to find out the route that the packets sent from your machine to a remote host, simply issue the following command:

```
[root@deep /]# traceroute www.redhat.com
traceroute to www.portal.redhat.com (206.132.41.202), 30 hops max, 38 byte packets
 1  portal.openna.com (207.253.108.5) 98.584 ms 1519.806 ms  109.911 ms
 2  fa5-1-0.rb02-piex.videotron.net (207.96.135.1) 149.888 ms 89.830 ms 109.914 ms
 3  ia-tlpt-bb01-fec1.videotron.net (207.253.253.53) 149.896 ms 99.873 ms 139.930 ms
 4  ia-cduc-bb02-ge2-0.videotron.net (207.253.253.61) 99.897 ms 169.863 ms 329.926 ms
 5  if-4-1.core1.Montreal.Teleglobe.net (207.45.204.5) 409.895 ms 1469.882 ms 109.902 ms
 6  if-1-1.core1.NewYork.Teleglobe.net (207.45.223.109) 189.920 ms 139.852 ms 109.939 ms
 7  206.132.150.133 (206.132.150.133) 99.902 ms 99.724 ms 119.914 ms
 8  pos1-0-2488M.wr2.CLE1.gblx.net (206.132.111.89) 189.899 ms 129.873 ms 129.934 ms
 9  pos8-0-2488m.kcy1.globalcenter.net (206.132.111.82) 169.890 ms 179.884 ms  169.933 ms
10  206.132.114.77 (206.132.114.77) 199.890 ms 179.771 ms 169.928 ms
11  pos8-0-2488M.wr2.SFO1.gblx.net (206.132.110.110) 159.909 ms 199.959 ms 179.837 ms
12  pos1-0-2488M.cr1.SNV2.gblx.net (208.48.118.118) 179.885 ms 309.855 ms 299.937 ms
13  pos0-0-0-155M.hr2.SNV2.gblx.net (206.132.151.46) 329.905 ms 179.843 ms 169.936 ms
14  206.132.41.202 (206.132.41.202)  2229.906 ms 199.752 ms 309.927 ms
```

Where <`www.redhat.com`> is the name or ip address of the host that you want to trace.

1.2 Display the number of times your Web pages have been accessed:
To display quickly the number of times your web page has been accessed use this command:

```
[root@deep /]# grep "GET / HTTP" /var/log/httpd/access_log | wc -l
      467
```

1.3 Shut down most services altogether:
As root, you can shut down most services altogether with the following command:

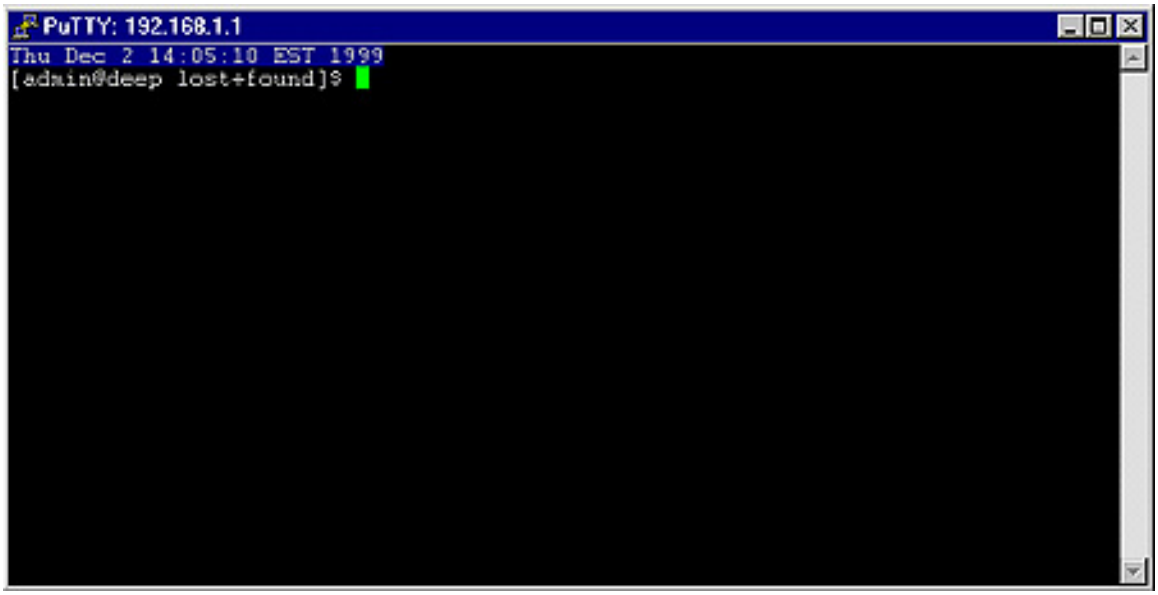
```
[root@deep /]# killall httpd smbd nmbd slapd named
```

The above command will shut down the Apache server, Samba services, LDAP server, and DNS server respectively.

1.4 Want a clock on the top of your terminal for all user?
Edit the **`profile`** file (`vi /etc/profile`) and add the following line:

```
PROMPT_COMMAND='echo -ne
"\0337\033[2;999r\033[1;1H\033[00;44m\033[K"`date`"\033[00m\0338"'
```

The result will look like:



1.5 Do you have `lsof` installed on your server?
If not, install it and execute `lsof -i`. This should list which ports you have open on your machine. The `lsof` program is a great tool as it will tell you which processes are listening on a given port.

```
[root@deep /]# lsof -i
COMMAND     PID     USER    FD  TYPE    DEVICE  SIZE    NODE    NAME
Inetd       344     root    4u          IPv4    327     TCP *:ssh (LISTEN)
```

1.6 Run commands on remote servers via `ssh` protocol without logging in:
The `ssh` command can also be used to run commands on remote systems without logging in. The output of the command is displayed, and control returns to the local system. Here is an example which will display all the users logged in on the remote system.

```
[admin@deep /]$ ssh boreas.openna.com who
admin@boreas.openna.com's password:
root     tty1     Dec  2 14:45
admin    tty2     Dec  2 14:45
wahib    pts/0    Dec  2 11:38
```

1.7 Filename Completion:
Tab filename completion allows you to type in portions of a filename or program, and then press `[TAB]`, and it will complete the filename for you. If there's more than one file or program that starts with what you already typed in, it will beep, and then when you press `[TAB]` again it will list all the files that start with what you initially typed.

NOTE: AFAIK, filename completion works only for `bash` by default but not for e.g. `ksh`. If you use `ksh` instead of `bash` as the command shell then to enable "Filename Completion" in `ksh`, you have to set the following:

```
set -o vi-tabcomplete
```

1.8 Special Characters:
You can quickly accomplish tasks that you perform frequently by using shortcut keys — one or more keys you press on the keyboard to complete a task. For example, special characters can be used on the Linux shell like the following:

`Control-d` : If you are in the shell and hit `control-d` you get logged off.

`Control-l`: If you are in the shell and hit `control-l` you clear the screen.

`?` : This is a wildcard. This can represent a single character. If you specified something at the command line like "`m?b`" Linux would look for `mob`, `mib`, `mub`, and every other letter/number between a-z, 0-9.

`*` : This can represent any number of characters. If you specified a "`mi*`" it would use `mit`, `mim`, `miiii`, `miya`, and ANYTHING that starts with "`mi`". "`m*l`" could by `mill`, `mull`, `ml`, and anything that starts with an "`m`" and ends with an "`l`".

`[]` - Specifies a range. if I did `m[o,u,i]m` Linux would think: `mim`, `mum`, `mom` if I did: `m[a-d]m` Linux would think: `mam`, `mbm`, `mcm`, `mdm`. Get the idea? The `[]`, `?`, and `*` are usually used with copying, deleting, and directory listings.

NOTE: EVERYTHING in Linux is CASE sensitive. This means "`Bill`" and "`bill`" are not the same thing. This allows for many files to be able to be stored, since "`Bill`" "`bill`" "`bIll`" "`biLl`", etc. can be different files. So, when using the `[]` stuff, you have to specify capital letters if any files you are dealing with have capital letters. Much of everything is lower case in UNIX, though.

1.9 Freeze a process ID temporally:
The UNIX `kill` command name is misleading: Only some incantations of the `kill` command actually terminate the target process. "`kill -STOP`" suspends the target process immediately and unconditionally. The process can still be resumed with "`kill -CONT`" as if nothing happened. This command can be useful when you want for example to freeze a suspicious process running on your system and conduct any further investigations at leisure.

```
[root@deep /]# kill -STOP 401
```

The above command will suspend the process ID 401, which is related to the `sshd` daemon on my running system. Of course the process number will be different on your server, therefore take this process number as an example only.

```
[root@deep /]# kill -CONT 401
```

The above command will resume the process ID 401, which is related to the `sshd` daemon on my running system.

APPENDIX B
Port list

Port list

The port numbers are divided into three ranges: the Well Known Ports, the Registered Ports, and the Dynamic and/or Private Ports. There are two series of ports, using two different protocols: TCP and UDP. They are different, although they can have the same port number. UDP ports can't be telneted. This appendix also includes a list of ports commonly used by Trojan horses. All open ports have a service or daemon running on it. A service or a daemon is nothing but the software running on these ports, which provide a certain service to the users who connect to it.

You can find out the corresponding services running on them, referring to the table below or to the RFC 1700 (http://www.cis.ohio-state.edu/rfc/), which contains the complete and updated list of Port Numbers and the corresponding popularly running services.

Well Known Ports:

The Well Known Ports are those from 0 through 1023 and are assigned by IANA (**I**nternet **A**ssigned **N**umbers **A**uthority). For the latest status, please check at: http://www.iana.org/

Keyword	Decimal	Description
	0/tcp	Reserved
	0/udp	Reserved
tcpmux	1/tcp	TCP Port Service
tcpmux	1/udp	TCP Port Service
compressnet	2/tcp	Management Utility
compressnet	2/udp	Management Utility
compressnet	3/tcp	Compression Process
compressnet	3/udp	Compression Process
#	4/tcp	Unassigned
#	4/udp	Unassigned
rje	5/tcp	Remote Job Entry
rje	5/udp	Remote Job Entry
#	6/tcp	Unassigned
#	6/udp	Unassigned
echo	7/tcp	Echo
echo	7/udp	Echo
#	8/tcp	Unassigned
#	8/udp	Unassigned
discard	9/tcp	Discard
discard	9/udp	Discard
#	10/tcp	Unassigned
#	10/udp	Unassigned
systat	11/tcp	Active Users
systat	11/udp	Active Users
#	12/tcp	Unassigned
#	12/udp	Unassigned
daytime	13/tcp	Daytime (RFC 867)
daytime	13/udp	Daytime (RFC 867)
#	14/tcp	Unassigned
#	14/udp	Unassigned
#	15/tcp	Unassigned
#	15/udp	Unassigned
#	16/tcp	Unassigned
#	16/udp	Unassigned
qotd	17/tcp	Quote of the Day
qotd	17/udp	Quote of the Day
msp	18/tcp	Message Send
msp	18/udp	Message Send
chargen	19/tcp	Character Generator
chargen	19/udp	Character Generator
ftp-data	20/tcp	File Transfer
ftp-data	20/udp	File Transfer
ftp	21/tcp	File Transfer
ftp	21/udp	File Transfer

Keyword	Decimal	Description
opc-job-track	424/tcp	IBM Operations
opc-job-track	424/udp	IBM Operations
icad-el	425/tcp	ICAD
icad-el	425/udp	ICAD
smartsdp	426/tcp	smartsdp
smartsdp	426/udp	smartsdp
svrloc	427/tcp	Server Location
svrloc	427/udp	Server Location
ocs_cmu	428/tcp	OCS_CMU
ocs_cmu	428/udp	OCS_CMU
ocs_amu	429/tcp	OCS_AMU
ocs_amu	429/udp	OCS_AMU
utmpsd	430/tcp	UTMPSD
utmpsd	430/udp	UTMPSD
utmpcd	431/tcp	UTMPCD
utmpcd	431/udp	UTMPCD
iasd	432/tcp	IASD
iasd	432/udp	IASD
nnsp	433/tcp	NNSP
nnsp	433/udp	NNSP
mobileip-agent	434/tcp	MobileIP-Agent
mobileip-agent	434/udp	MobileIP-Agent
mobilip-mn	435/tcp	MobilIP-MN
mobilip-mn	435/udp	MobilIP-MN
dna-cml	436/tcp	DNA-CML
dna-cml	436/udp	DNA-CML
comscm	437/tcp	comscm
comscm	437/udp	comscm
dsfgw	438/tcp	dsfgw
dsfgw	438/udp	dsfgw
dasp	439/tcp	dasp
dasp	439/udp	dasp
sgcp	440/tcp	sgcp
sgcp	440/udp	sgcp
decvms-sysmgt	441/tcp	decvms-sysmgt
decvms-sysmgt	441/udp	decvms-sysmgt
cvc_hostd	442/tcp	cvc_hostd
cvc_hostd	442/udp	cvc_hostd
https	443/tcp	http proto TLS/SSL
https	443/udp	http proto TLS/SSL
snpp	444/tcp	Simple Network
snpp	444/udp	Simple Network
microsoft-ds	445/tcp	Microsoft-DS
microsoft-ds	445/udp	Microsoft-DS

```
ssh             22/tcp    SSH Remote Login
ssh             22/udp    SSH Remote Login
telnet          23/tcp    Telnet
telnet          23/udp    Telnet
#               24/tcp    any private mail sys
#               24/udp    any private mail sys
smtp            25/tcp    Simple Mail Transfer
smtp            25/udp    Simple Mail Transfer
#               26/tcp    Unassigned
#               26/udp    Unassigned
nsw-fe          27/tcp    NSW User System FE
nsw-fe          27/udp    NSW User System FE
#               28/tcp    Unassigned
#               28/udp    Unassigned
msg-icp         29/tcp    MSG ICP
msg-icp         29/udp    MSG ICP
#               30/tcp    Unassigned
#               30/udp    Unassigned
msg-auth        31/tcp    MSG Authentication
msg-auth        31/udp    MSG Authentication
#               32/tcp    Unassigned
#               32/udp    Unassigned
dsp             33/tcp    Display Support
dsp             33/udp    Display Support
#               34/tcp    Unassigned
#               34/udp    Unassigned
#               35/tcp    any private printer
#               35/udp    any private printer
#               36/tcp    Unassigned
#               36/udp    Unassigned
time            37/tcp    Time
time            37/udp    Time
rap             38/tcp    Route Access
rap             38/udp    Route Access
rlp             39/tcp    Resource Location
rlp             39/udp    Resource Location
#               40/tcp    Unassigned
#               40/udp    Unassigned
graphics        41/tcp    Graphics
graphics        41/udp    Graphics
nameserver      42/tcp    Host Name Server
nameserver      42/udp    Host Name Server
nicname         43/tcp    Who Is
nicname         43/udp    Who Is
mpm-flags       44/tcp    MPM FLAGS Protocol
mpm-flags       44/udp    MPM FLAGS Protocol
mpm             45/tcp    MPM [recv]
mpm             45/udp    MPM [recv]
mpm-snd         46/tcp    MPM [default send]
mpm-snd         46/udp    MPM [default send]
ni-ftp          47/tcp    NI FTP
ni-ftp          47/udp    NI FTP
auditd          48/tcp    Digital Audit Daemon
auditd          48/udp    Digital Audit Daemon
tacacs          49/tcp    Login Host Protocol
tacacs          49/udp    Login Host Protocol
re-mail-ck      50/tcp    Remote Mail Checking
re-mail-ck      50/udp    Remote Mail Checking
la-maint        51/tcp    IMP
la-maint        51/udp    IMP
xns-time        52/tcp    XNS Time Protocol
xns-time        52/udp    XNS Time Protocol
domain          53/tcp    Domain Name Server
domain          53/udp    Domain Name Server
xns-ch          54/tcp    XNS Clearinghouse
xns-ch          54/udp    XNS Clearinghouse
isi-gl          55/tcp    ISI Graphics Lang
isi-gl          55/udp    ISI Graphics Lang
xns-auth        56/tcp    XNS Authentication
xns-auth        56/udp    XNS Authentication
#               57/tcp    Private term access
ddm-rdb         446/tcp    DDM-RDB
ddm-rdb         446/udp    DDM-RDB
ddm-dfm         447/tcp    DDM-RFM
ddm-dfm         447/udp    DDM-RFM
ddm-ssl         448/tcp    DDM-SSL
ddm-ssl         448/udp    DDM-SSL
as-servermap    449/tcp    AS Server Mapper
as-servermap    449/udp    AS Server Mapper
tserver         450/tcp    TServer
tserver         450/udp    TServer
sfs-smp-net     451/tcp    Cray Network
sfs-smp-net     451/udp    Cray Network
sfs-config      452/tcp    Cray SFS config
sfs-config      452/udp    Cray SFS config
creativeserver  453/tcp    CreativeServer
creativeserver  453/udp    CreativeServer
contentserver   454/tcp    ContentServer
contentserver   454/udp    ContentServer
creativepartnr  455/tcp    CreativePartnr
creativepartnr  455/udp    CreativePartnr
macon-tcp       456/tcp    macon-tcp
macon-udp       456/udp    macon-udp
scohelp         457/tcp    scohelp
scohelp         457/udp    scohelp
appleqtc        458/tcp    apple quick time
appleqtc        458/udp    apple quick time
ampr-rcmd       459/tcp    ampr-rcmd
ampr-rcmd       459/udp    ampr-rcmd
skronk          460/tcp    skronk
skronk          460/udp    skronk
datasurfsrv     461/tcp    DataRampSrv
datasurfsrv     461/udp    DataRampSrv
datasurfsrvsec  462/tcp    DataRampSrvSec
datasurfsrvsec  462/udp    DataRampSrvSec
alpes           463/tcp    alpes
alpes           463/udp    alpes
kpasswd         464/tcp    kpasswd
kpasswd         464/udp    kpasswd
#               465        Unassigned
digital-vrc     466/tcp    digital-vrc
digital-vrc     466/udp    digital-vrc
mylex-mapd      467/tcp    mylex-mapd
mylex-mapd      467/udp    mylex-mapd
photuris        468/tcp    proturis
photuris        468/udp    proturis
rcp             469/tcp    Radio Control Proto
rcp             469/udp    Radio Control Proto
scx-proxy       470/tcp    scx-proxy
mondex          471/tcp    Mondex
mondex          471/udp    Mondex
ljk-login       472/tcp    ljk-login
ljk-login       472/udp    ljk-login
hybrid-pop      473/tcp    hybrid-pop
hybrid-pop      473/udp    hybrid-pop
tn-tl-w1        474/tcp    tn-tl-w1
tn-tl-w2        474/udp    tn-tl-w2
tcpnethaspsrv   475/tcp    tcpnethaspsrv
tcpnethaspsrv   475/udp    tcpnethaspsrv
tn-tl-fd1       476/tcp    tn-tl-fd1
tn-tl-fd1       476/udp    tn-tl-fd1
ss7ns           477/tcp    ss7ns
ss7ns           477/udp    ss7ns
spsc            478/tcp    spsc
spsc            478/udp    spsc
iafserver       479/tcp    iafserver
iafserver       479/udp    iafserver
iafdbase        480/tcp    iafdbase
iafdbase        480/udp    iafdbase
ph              481/tcp    Ph service
ph              481/udp    Ph service
bgs-nsi         482/tcp    bgs-nsi
```

```
#               57/udp    Private term access
xns-mail        58/tcp    XNS Mail
xns-mail        58/udp    XNS Mail
#               59/tcp    Private file service
#               59/udp    Private file service
#               60/tcp    Unassigned
#               60/udp    Unassigned
ni-mail         61/tcp    NI MAIL
ni-mail         61/udp    NI MAIL
acas            62/tcp    ACA Services
acas            62/udp    ACA Services
whois++         63/tcp    whois++
whois++         63/udp    whois++
covia           64/tcp    Com Integrator (CI)
covia           64/udp    Com Integrator (CI)
tacacs-ds       65/tcp    TACACS-Database Serv
tacacs-ds       65/udp    TACACS-Database Serv
sql*net         66/tcp    Oracle SQL*NET
sql*net         66/udp    Oracle SQL*NET
bootps          67/tcp    Bootstrap Server
bootps          67/udp    Bootstrap Server
bootpc          68/tcp    Bootstrap Client
bootpc          68/udp    Bootstrap Client
tftp            69/tcp    Trivial File Trans
tftp            69/udp    Trivial File Trans
gopher          70/tcp    Gopher
gopher          70/udp    Gopher
netrjs-1        71/tcp    Remote Job Service
netrjs-1        71/udp    Remote Job Service
netrjs-2        72/tcp    Remote Job Service
netrjs-2        72/udp    Remote Job Service
netrjs-3        73/tcp    Remote Job Service
netrjs-3        73/udp    Remote Job Service
netrjs-4        74/tcp    Remote Job Service
netrjs-4        74/udp    Remote Job Service
#               75/tcp    Private dial out
#               75/udp    Private dial out
deos            76/tcp    ExternalObject Store
deos            76/udp    ExternalObject Store
#               77/tcp    Private RJE service
#               77/udp    Private RJE service
vettcp          78/tcp    vettcp
vettcp          78/udp    vettcp
finger          79/tcp    Finger
finger          79/udp    Finger
http            80/tcp    World Wide Web HTTP
http            80/udp    World Wide Web HTTP
hosts2-ns       81/tcp    HOSTS2 Name Server
hosts2-ns       81/udp    HOSTS2 Name Server
xfer            82/tcp    XFER Utility
xfer            82/udp    XFER Utility
mit-ml-dev      83/tcp    MIT ML Device
mit-ml-dev      83/udp    MIT ML Device
ctf             84/tcp    CommonTrace Facility
ctf             84/udp    CommonTrace Facility
mit-ml-dev      85/tcp    MIT ML Device
mit-ml-dev      85/udp    MIT ML Device
mfcobol         86/tcp    Micro Focus Cobol
mfcobol         86/udp    Micro Focus Cobol
#               87/tcp    Private term link
#               87/udp    Private term link
kerberos        88/tcp    Kerberos
kerberos        88/udp    Kerberos
su-mit-tg       89/tcp    SU/MITTelnet Gateway
su-mit-tg       89/udp    SU/MITTelnet Gateway
dnsix           90/tcp    DNSIX
dnsix           90/udp    DNSIX
mit-dov         91/tcp    MIT Dover Spooler
mit-dov         91/udp    MIT Dover Spooler
npp             92/tcp    Network Printing
npp             92/udp    Network Printing
bgs-nsi         482/udp   bgs-nsi
ulpnet          483/tcp   ulpnet
ulpnet          483/udp   ulpnet
integra-sme     484/tcp   Integra Software
integra-sme     484/udp   Integra Software
powerburst      485/tcp   Air Soft Power Burst
powerburst      485/udp   Air Soft Power Burst
avian           486/tcp   avian
avian           486/udp   avian
saft            487/tcp   saft
saft            487/udp   saft
gss-http        488/tcp   gss-http
gss-http        488/udp   gss-http
nest-protocol   489/tcp   nest-protocol
nest-protocol   489/udp   nest-protocol
micom-pfs       490/tcp   micom-pfs
micom-pfs       490/udp   micom-pfs
go-login        491/tcp   go-login
go-login        491/udp   go-login
ticf-1          492/tcp   Transport for FNA
ticf-1          492/udp   Transport for FNA
ticf-2          493/tcp   Transport for FNA
ticf-2          493/udp   Transport for FNA
pov-ray         494/tcp   POV-Ray
pov-ray         494/udp   POV-Ray
intecourier     495/tcp   intecourier
intecourier     495/udp   intecourier
pim-rp-disc     496/tcp   PIM-RP-DISC
pim-rp-disc     496/udp   PIM-RP-DIS
dantz           497/tcp   dantz
dantz           497/udp   dantz
siam            498/tcp   siam
siam            498/udp   siam
iso-ill         499/tcp   ISO ILL Protocol
iso-ill         499/udp   ISO ILL Protocol
isakmp          500/tcp   isakmp
isakmp          500/udp   isakmp
stmf            501/tcp   STMF
stmf            501/udp   STMF
asa-appl-proto  502/tcp   asa-appl-proto
asa-appl-proto  502/udp   asa-appl-proto
intrinsa        503/tcp   Intrinsa
intrinsa        503/udp   Intrinsa
citadel         504/tcp   citadel
citadel         504/udp   citadel
mailbox-lm      505/tcp   mailbox-lm
mailbox-lm      505/udp   mailbox-lm
ohimsrv         506/tcp   ohimsrv
ohimsrv         506/udp   ohimsrv
crs             507/tcp   crs
crs             507/udp   crs
xvttp           508/tcp   xvttp
xvttp           508/udp   xvttp
snare           509/tcp   snare
snare           509/udp   snare
fcp             510/tcp   FirstClass Protocol
fcp             510/udp   FirstClass Protocol
passgo          511/tcp   PassGo
passgo          511/udp   PassGo
exec            512/tcp   remote process exec
comsat          512/udp
biff            512/udp   used by mail system
login           513/tcp   remote login
who             513/udp   maintains data bases
shell           514/tcp   cmd
syslog          514/udp
printer         515/tcp   spooler
printer         515/udp   spooler
videotex        516/tcp   videotex
videotex        516/udp   videotex
talk            517/tcp   like tenex
```

```
dcp             93/tcp    Device Control
dcp             93/udp    Device Control
objcall         94/tcp    Tivoli Object
objcall         94/udp    Tivoli Object
supdup          95/tcp    SUPDUP
supdup          95/udp    SUPDUP
dixie           96/tcp    DIXIE Specification
dixie           96/udp    DIXIE Specification
swift-rvf       97/tcp    Swift Remote
swift-rvf       97/udp    Swift Remote
tacnews         98/tcp    TAC News
tacnews         98/udp    TAC News
metagram        99/tcp    Metagram Relay
metagram        99/udp    Metagram Relay
newacct         100/tcp   [unauthorized use]
hostname        101/tcp   NIC Host Name Server
hostname        101/udp   NIC Host Name Server
iso-tsap        102/tcp   ISO-TSAP Class 0
iso-tsap        102/udp   ISO-TSAP Class 0
gppitnp         103/tcp   Genesis Trans Net
gppitnp         103/udp   Genesis Trans Net
acr-nema        104/tcp   ACR-NEMA Digital
acr-nema        104/udp   ACR-NEMA Digital
cso             105/tcp   CCSO name server
cso             105/udp   CCSO name server
csnet-ns        105/tcp   Mailbox Nameserver
csnet-ns        105/udp   Mailbox Nameserver
3com-tsmux      106/tcp   3COM-TSMUX
3com-tsmux      106/udp   3COM-TSMUX
rtelnet         107/tcp   Remote Telnet
rtelnet         107/udp   Remote Telnet
snagas          108/tcp   SNA
snagas          108/udp   SNA
pop2            109/tcp   Post Office - V2
pop2            109/udp   Post Office - V2
pop3            110/tcp   Post Office - V3
pop3            110/udp   Post Office - V3
sunrpc          111/tcp   SUN Remote Proc Call
sunrpc          111/udp   SUN Remote Proc Call
mcidas          112/tcp   McIDAS
mcidas          112/udp   McIDAS
ident           113/tcp
auth            113/tcp   Auth Service
auth            113/udp   Auth Service
audionews       114/tcp   Audio News Multicast
audionews       114/udp   Audio News Multicast
sftp            115/tcp   Simple FTP
sftp            115/udp   Simple FTP
ansanotify      116/tcp   ANSA REX Notify
ansanotify      116/udp   ANSA REX Notify
uucp-path       117/tcp   UUCP Path Service
uucp-path       117/udp   UUCP Path Service
sqlserv         118/tcp   SQL Services
sqlserv         118/udp   SQL Services
nntp            119/tcp   NNTP
nntp            119/udp   NNTP
cfdptkt         120/tcp   CFDPTKT
cfdptkt         120/udp   CFDPTKT
erpc            121/tcp   Remote Pro.Call
erpc            121/udp   Remote Pro.Call
smakynet        122/tcp   SMAKYNET
smakynet        122/udp   SMAKYNET
ntp             123/tcp   Network Time Proto
ntp             123/udp   Network Time Proto
ansatrader      124/tcp   ANSA REX Trader
ansatrader      124/udp   ANSA REX Trader
locus-map       125/tcp   Locus Net Map Ser
locus-map       125/udp   Locus Net Map Ser
nxedit          126/tcp   NXEdit
nxedit          126/udp   NXEdit
#unitary        126/tcp   Unisys Unitary Login
talk            517/udp   like tenex
ntalk           518/tcp
ntalk           518/udp
utime           519/tcp   unixtime
utime           519/udp   unixtime
efs             520/tcp   extended file name
router          520/udp   routing process
ripng           521/tcp   ripng
ripng           521/udp   ripng
ulp             522/tcp   ULP
ulp             522/udp   ULP
ibm-db2         523/tcp   IBM-DB2
ibm-db2         523/udp   IBM-DB2
ncp             524/tcp   NCP
ncp             524/udp   NCP
timed           525/tcp   Timeserver
timed           525/udp   Timeserver
tempo           526/tcp   Newdate
tempo           526/udp   Newdate
stx             527/tcp   Stock IXChange
stx             527/udp   Stock IXChange
custix          528/tcp   Customer IXChange
custix          528/udp   Customer IXChange
irc-serv        529/tcp   IRC-SERV
irc-serv        529/udp   IRC-SERV
courier         530/tcp   rpc
courier         530/udp   rpc
conference      531/tcp   chat
conference      531/udp   chat
netnews         532/tcp   readnews
netnews         532/udp   readnews
netwall         533/tcp   Emergency broadcasts
netwall         533/udp   Emergency broadcasts
mm-admin        534/tcp   MegaMedia Admin
mm-admin        534/udp   MegaMedia Admin
iiop            535/tcp   iiop
iiop            535/udp   iiop
opalis-rdv      536/tcp   opalis-rdv
opalis-rdv      536/udp   opalis-rdv
nmsp            537/tcp   Media Streaming
nmsp            537/udp   Media Streaming
gdomap          538/tcp   gdomap
gdomap          538/udp   gdomap
apertus-ldp     539/tcp   Apertus Technologies
apertus-ldp     539/udp   Apertus Technologies
uucp            540/tcp   uucpd
uucp            540/udp   uucpd
uucp-rlogin     541/tcp   uucp-rlogin
uucp-rlogin     541/udp   uucp-rlogin
commerce        542/tcp   commerce
commerce        542/udp   commerce
klogin          543/tcp
klogin          543/udp
kshell          544/tcp   krcmd
kshell          544/udp   krcmd
appleqtcsrvr    545/tcp   appleqtcsrvr
appleqtcsrvr    545/udp   appleqtcsrvr
dhcpv6-client   546/tcp   DHCPv6 Client
dhcpv6-client   546/udp   DHCPv6 Client
dhcpv6-server   547/tcp   DHCPv6 Server
dhcpv6-server   547/udp   DHCPv6 Server
afpovertcp      548/tcp   AFP over TCP
afpovertcp      548/udp   AFP over TCP
idfp            549/tcp   IDFP
idfp            549/udp   IDFP
new-rwho        550/tcp   new-who
new-rwho        550/udp   new-who
cybercash       551/tcp   cybercash
cybercash       551/udp   cybercash
deviceshare     552/tcp   deviceshare
deviceshare     552/udp   deviceshare
```

```
#unitary      126/udp   Unisys Unitary Login
locus-con     127/tcp   Locus Conn Server
locus-con     127/udp   Locus Conn Server
gss-xlicen    128/tcp   GSS X Verification
gss-xlicen    128/udp   GSS X Verification
pwdgen        129/tcp   Password Generator
pwdgen        129/udp   Password Generator
cisco-fna     130/tcp   cisco FNATIVE
cisco-fna     130/udp   cisco FNATIVE
cisco-tna     131/tcp   cisco TNATIVE
cisco-tna     131/udp   cisco TNATIVE
cisco-sys     132/tcp   cisco SYSMAINT
cisco-sys     132/udp   cisco SYSMAINT
statsrv       133/tcp   Statistics Service
statsrv       133/udp   Statistics Service
ingres-net    134/tcp   INGRES-NET Service
ingres-net    134/udp   INGRES-NET Service
epmap         135/tcp   DCE
epmap         135/udp   DCE
profile       136/tcp   PROFILE Naming Sys
profile       136/udp   PROFILE Naming Sys
netbios-ns    137/tcp   NETBIOS Name Serv
netbios-ns    137/udp   NETBIOS Name Serv
netbios-dgm   138/tcp   NETBIOS Data Serv
netbios-dgm   138/udp   NETBIOS Data Serv
netbios-ssn   139/tcp   NETBIOS Session Serv
netbios-ssn   139/udp   NETBIOS Session Serv
emfis-data    140/tcp   EMFIS Data Serv
emfis-data    140/udp   EMFIS Data Serv
emfis-cntl    141/tcp   EMFIS Control Serv
emfis-cntl    141/udp   EMFIS Control Serv
bl-idm        142/tcp   Britton-Lee IDM
bl-idm        142/udp   Britton-Lee IDM
imap          143/tcp   IMAP Protocol
imap          143/udp   IMAP Protocol
uma           144/tcp   UMA Protocol
uma           144/udp   UMA Protocol
uaac          145/tcp   UAAC Protocol
uaac          145/udp   UAAC Protocol
iso-tp0       146/tcp   ISO-IP0
iso-tp0       146/udp   ISO-IP0
iso-ip        147/tcp   ISO-IP
iso-ip        147/udp   ISO-IP
jargon        148/tcp   Jargon
jargon        148/udp   Jargon
aed-512       149/tcp   AED 512 Emulation
aed-512       149/udp   AED 512 Emulation
sql-net       150/tcp   SQL-NET
sql-net       150/udp   SQL-NET
hems          151/tcp   HEMS
hems          151/udp   HEMS
bftp          152/tcp   Background FTP
bftp          152/udp   Background FTP
sgmp          153/tcp   SGMP
sgmp          153/udp   SGMP
netsc-prod    154/tcp   NETSC
netsc-prod    154/udp   NETSC
netsc-dev     155/tcp   NETSC
netsc-dev     155/udp   NETSC
sqlsrv        156/tcp   SQL Service
sqlsrv        156/udp   SQL Service
knet-cmp      157/tcp   KNET/VM Protocol
knet-cmp      157/udp   KNET/VM Protocol
pcmail-srv    158/tcp   PCMail Server
pcmail-srv    158/udp   PCMail Server
nss-routing   159/tcp   NSS-Routing
nss-routing   159/udp   NSS-Routing
sgmp-traps    160/tcp   SGMP-TRAPS
sgmp-traps    160/udp   SGMP-TRAPS
snmp          161/tcp   SNMP
snmp          161/udp   SNMP
pirp          553/tcp    pirp
pirp          553/udp    pirp
rtsp          554/tcp    Real Time Stream
rtsp          554/udp    Real Time Strea
dsf           555/tcp
dsf           555/udp
remotefs      556/tcp    rfs server
remotefs      556/udp    rfs server
openvms-sysipc  557/tcp    openvms-sysipc
openvms-sysipc  557/udp    openvms-sysipc
sdnskmp       558/tcp    SDNSKMP
sdnskmp       558/udp    SDNSKMP
teedtap       559/tcp    TEEDTAP
teedtap       559/udp    TEEDTAP
rmonitor      560/tcp    rmonitord
rmonitor      560/udp    rmonitord
monitor       561/tcp
monitor       561/udp
chshell       562/tcp    chcmd
chshell       562/udp    chcmd
nntps         563/tcp    nntp over TLS/SSL
nntps         563/udp    nntp over TLS/SSL
9pfs          564/tcp    plan 9 file service
9pfs          564/udp    plan 9 file service
whoami        565/tcp    whoami
whoami        565/udp    whoami
streettalk    566/tcp    streettalk
streettalk    566/udp    streettalk
banyan-rpc    567/tcp    banyan-rpc
banyan-rpc    567/udp    banyan-rpc
ms-shuttle    568/tcp    microsoft shuttle
ms-shuttle    568/udp    microsoft shuttle
ms-rome       569/tcp    microsoft rome
ms-rome       569/udp    microsoft rome
meter         570/tcp    demon
meter         570/udp    demon
meter         571/tcp    udemon
meter         571/udp    udemon
sonar         572/tcp    sonar
sonar         572/udp    sonar
banyan-vip    573/tcp    banyan-vip
banyan-vip    573/udp    banyan-vip
ftp-agent     574/tcp    FTP Software Agent
ftp-agent     574/udp    FTP Software Agent
vemmi         575/tcp    VEMMI
vemmi         575/udp    VEMMI
ipcd          576/tcp    ipcd
ipcd          576/udp    ipcd
vnas          577/tcp    vnas
vnas          577/udp    vnas
ipdd          578/tcp    ipdd
ipdd          578/udp    ipdd
decbsrv       579/tcp    decbsrv
decbsrv       579/udp    decbsrv
sntp-heartbeat  580/tcp    SNTP HEARTBEAT
sntp-heartbeat  580/udp    SNTP HEARTBEAT
bdp           581/tcp    Bundle Discovery
bdp           581/udp    Bundle Discovery
scc-security    582/tcp    SCC Security
scc-security    582/udp    SCC Security
philips-vc    583/tcp    Philips Video
philips-vc    583/udp    Philips Video
keyserver     584/tcp    Key Server
keyserver     584/udp    Key Server
imap4-ssl     585/tcp    IMAP4+SSL
imap4-ssl     585/udp    IMAP4+SSL
password-chg    586/tcp    Password Change
password-chg    586/udp    Password Change
submission      587/tcp    Submission
submission      587/udp    Submission
cal           588/tcp    CAL
```

```
snmptrap      162/tcp   SNMPTRAP
snmptrap      162/udp   SNMPTRAP
cmip-man      163/tcp   CMIP/TCP Manager
cmip-man      163/udp   CMIP/TCP Manager
cmip-agent    164/tcp   CMIP/TCP Agent
smip-agent    164/udp   CMIP/TCP Agent
xns-courier   165/tcp   Xerox
xns-courier   165/udp   Xerox
s-net         166/tcp   Sirius Systems
s-net         166/udp   Sirius Systems
namp          167/tcp   NAMP
namp          167/udp   NAMP
rsvd          168/tcp   RSVD
rsvd          168/udp   RSVD
send          169/tcp   SEND
send          169/udp   SEND
print-srv     170/tcp   Network PostScript
print-srv     170/udp   Network PostScript
multiplex     171/tcp   Network Innovations
multiplex     171/udp   Network Innovations
cl/1          172/tcp   Network Innovations
cl/1          172/udp   Network Innovations
xyplex-mux    173/tcp   Xyplex
xyplex-mux    173/udp   Xyplex
mailq         174/tcp   MAILQ
mailq         174/udp   MAILQ
vmnet         175/tcp   VMNET
vmnet         175/udp   VMNET
genrad-mux    176/tcp   GENRAD-MUX
genrad-mux    176/udp   GENRAD-MUX
xdmcp         177/tcp   X Display Manager
xdmcp         177/udp   X Display Manager
nextstep      178/tcp   NextStep Win Server
nextstep      178/udp   NextStep Win Server
bgp           179/tcp   Border Gateway
bgp           179/udp   Border Gateway
ris           180/tcp   Intergraph
ris           180/udp   Intergraph
unify         181/tcp   Unify
unify         181/udp   Unify
audit         182/tcp   Unisys Audit SITP
audit         182/udp   Unisys Audit SITP
ocbinder      183/tcp   OCBinder
ocbinder      183/udp   OCBinder
ocserver      184/tcp   OCServer
ocserver      184/udp   OCServer
remote-kis    185/tcp   Remote-KIS
remote-kis    185/udp   Remote-KIS
kis           186/tcp   KIS Protocol
kis           186/udp   KIS Protocol
aci           187/tcp   App Communication
aci           187/udp   App Communication
mumps         188/tcp   Plus Five's MUMPS
mumps         188/udp   Plus Five's MUMPS
qft           189/tcp   Queued File Trans
qft           189/udp   Queued File Trans
gacp          190/tcp   Gateway Acc Control
gacp          190/udp   Gateway Acc Control
prospero      191/tcp   Prospero Directory
prospero      191/udp   Prospero Directory
osu-nms       192/tcp   Net Monitoring Sys
osu-nms       192/udp   Net Monitoring Sys
srmp          193/tcp   Spider Monitoring
srmp          193/udp   Spider Monitoring
irc           194/tcp   Internet Relay Chat
irc           194/udp   Internet Relay Chat
dn6-nlm-aud   195/tcp   DNSIX Module Audit
dn6-nlm-aud   195/udp   DNSIX Module Audit
dn6-smm-red   196/tcp   DNSIX Session Mgt
dn6-smm-red   196/udp   DNSIX Session Mgt
dls           197/tcp   Directory Location
cal           588/udp    CAL
eyelink       589/tcp    EyeLink
eyelink       589/udp    EyeLink
tns-cml       590/tcp    TNS CML
tns-cml       590/udp    TNS CML
http-alt      591/tcp    FileMaker
http-alt      591/udp    FileMaker
eudora-set    592/tcp    Eudora Set
eudora-set    592/udp    Eudora Set
http-rpc-epmap  593/tcp    HTTP RPC Ep Map
http-rpc-epmap  593/udp    HTTP RPC Ep Map
tpip          594/tcp    TPIP
tpip          594/udp    TPIP
cab-protocol    595/tcp    CAB Protocol
cab-protocol    595/udp    CAB Protocol
smsd          596/tcp    SMSD
smsd          596/udp    SMSD
ptcnameservice  597/tcp    PTC Name Service
ptcnameservice  597/udp    PTC Name Service
sco-websrvrmg3  598/tcp    SCO Web Server
sco-websrvrmg3  598/udp    SCO Web Server
acp           599/tcp    Aeolon Core Protocol
acp           599/udp    Aeolon Core Protocol
ipcserver     600/tcp    Sun IPC server
ipcserver     600/udp    Sun IPC server
#             601-605    Unassigned
urm           606/tcp    Cray Unified
urm           606/udp    Cray Unified
nqs           607/tcp    nqs
nqs           607/udp    nqs
sift-uft      608/tcp    sender Init/Unsolici
sift-uft      608/udp    Sender-Init/Unsolici
npmp-trap     609/tcp    npmp-trap
npmp-trap     609/udp    npmp-trap
npmp-local    610/tcp    npmp-local
npmp-local    610/udp    npmp-local
npmp-gui      611/tcp    npmp-gui
npmp-gui      611/udp    npmp-gui
hmmp-ind      612/tcp    HMMP Indication
hmmp-ind      612/udp    HMMP Indication
hmmp-op       613/tcp    HMMP Operation
hmmp-op       613/udp    HMMP Operation
sshell        614/tcp    SSLshell
sshell        614/udp    SSLshell
sco-inetmgr   615/tcp    Internet Config Man
sco-inetmgr   615/udp    Internet Config Man
sco-sysmgr    616/tcp    SCO System Admin
sco-sysmgr    616/udp    SCO System Admin
sco-dtmgr     617/tcp    SCO Desktop Admin
sco-dtmgr     617/udp    SCO Desktop Admin
dei-icda      618/tcp    DEI-ICDA
dei-icda      618/udp    DEI-ICDA
digital-evm   619/tcp    Digital EVM
digital-evm   619/udp    Digital EVM
sco-websrvrmgr  620/tcp    SCO WebServer
sco-websrvrmgr  620/udp    SCO WebServer
escp-ip       621/tcp    ESCP
escp-ip       621/udp    ESCP
collaborator    622/tcp    Collaborator
collaborator    622/udp    Collaborator
aux_bus_shunt   623/tcp    Aux Bus Shunt
aux_bus_shunt   623/udp    Aux Bus Shunt
cryptoadmin     624/tcp    Crypto Admin
cryptoadmin     624/udp    Crypto Admin
dec_dlm     625/tcp    DEC DLM
dec_dlm     625/udp    DEC DLM
asia        626/tcp    ASIA
asia        626/udp    ASIA
passgo-tivoli   627/tcp    PassGo Tivoli
passgo-tivoli   627/udp    PassGo Tivoli
qmqp        628/tcp    QMQP
```

```
dls             197/udp   Directory Location
dls-mon         198/tcp   Directory Location
dls-mon         198/udp   Directory Location
smux            199/tcp   SMUX
smux            199/udp   SMUX
src             200/tcp   IBM System Resource
src             200/udp   IBM System Resource
at-rtmp         201/tcp   AppleTalk Routing
at-rtmp         201/udp   AppleTalk Routing
at-nbp          202/tcp   AppleTalk Name
at-nbp          202/udp   AppleTalk Name
at-3            203/tcp   AppleTalk Unused
at-3            203/udp   AppleTalk Unused
at-echo         204/tcp   AppleTalk Echo
at-echo         204/udp   AppleTalk Echo
at-5            205/tcp   AppleTalk Unused
at-5            205/udp   AppleTalk Unused
at-zis          206/tcp   AppleTalk Zone
at-zis          206/udp   AppleTalk Zone
at-7            207/tcp   AppleTalk Unused
at-7            207/udp   AppleTalk Unused
at-8            208/tcp   AppleTalk Unused
at-8            208/udp   AppleTalk Unused
qmtp            209/tcp   Quick Mail Transfer
qmtp            209/udp   Quick Mail Transfer
z39.50          210/tcp   ANSI Z39.50
z39.50          210/udp   ANSI Z39.50
914c/g          211/tcp   Texas Instruments
914c/g          211/udp   Texas Instruments
anet            212/tcp   ATEXSSTR
anet            212/udp   ATEXSSTR
ipx             213/tcp   IPX
ipx             213/udp   IPX
vmpwscs         214/tcp   VM PWSCS
vmpwscs         214/udp   VM PWSCS
softpc          215/tcp   Insignia Solutions
softpc          215/udp   Insignia Solutions
CAIlic          216/tcp   Computer Associates
CAIlic          216/udp   Computer Associates
dbase           217/tcp   dBASE Unix
dbase           217/udp   dBASE Unix
mpp             218/tcp   Netix Message Post
mpp             218/udp   Netix Message Post
uarps           219/tcp   Unisys ARPs
uarps           219/udp   Unisys ARPs
imap3           220/tcp   IMAP v3
imap3           220/udp   IMAP v3
fln-spx         221/tcp   Berkeley rlogind
fln-spx         221/udp   Berkeley rlogind
rsh-spx         222/tcp   Berkeley rshd
rsh-spx         222/udp   Berkeley rshd
cdc             223/tcp   Certificate Distrib
cdc             223/udp   Certificate Distrib
masqdialer      224/tcp   masqdialer
masqdialer      224/udp   masqdialer
#               225-241   Reserved
direct          242/tcp   Direct
direct          242/udp   Direct
sur-meas        243/tcp   Survey Measurement
sur-meas        243/udp   Survey Measurement
inbusiness      244/tcp   inbusiness
inbusiness      244/udp   inbusiness
link            245/tcp   LINK
link            245/udp   LINK
dsp3270         246/tcp   Display Systems
dsp3270         246/udp   Display Systems
subntbcst_tftp  247/tcp    SUBNTBCST_TFTP
subntbcst_tftp  247/udp    SUBNTBCST_TFTP
bhfhs           248/tcp   bhfhs
bhfhs           248/udp   bhfhs
#               249-255   Reserved
qmqp            628/udp    QMQP
3com-amp3       629/tcp    3Com AMP3
3com-amp3       629/udp    3Com AMP3
rda             630/tcp    RDA
rda             630/udp    RDA
ipp             631/tcp    Internet Printing
ipp             631/udp    Internet Printing
bmpp            632/tcp    bmpp
bmpp            632/udp    bmpp
servstat        633/tcp    Service Status update
servstat        633/udp    Service Status update
ginad           634/tcp    ginad
ginad           634/udp    ginad
rlzdbase        635/tcp    RLZ DBase
rlzdbase        635/udp    RLZ DBase
ldaps           636/tcp    ldap protocol TLS/SSL
ldaps           636/udp    ldap protocol TLS/SSL
lanserver       637/tcp    lanserver
lanserver       637/udp    lanserver
mcns-sec        638/tcp    mcns-sec
mcns-sec        638/udp    mcns-sec
msdp            639/tcp    MSDP
msdp            639/udp    MSDP
entrust-sps     640/tcp    entrust-sps
entrust-sps     640/udp    entrust-sps
repcmd          641/tcp    repcmd
repcmd          641/udp    repcmd
esro-emsdp      642/tcp    ESRO-EMSDP V1.3
esro-emsdp      642/udp    ESRO-EMSDP V1.3
sanity          643/tcp    SANity
sanity          643/udp    SANity
dwr             644/tcp    dwr
dwr             644/udp    dwr
pssc            645/tcp    PSSC
pssc            645/udp    PSSC
ldp             646/tcp    LDP
ldp             646/udp    LDP
dhcp-failover   647/tcp    DHCP Failover
dhcp-failover   647/udp    DHCP Failover
rrp             648/tcp    Registry Registrar
rrp             648/udp    Registry Registrar
aminet          649/tcp    Aminet
aminet          649/udp    Aminet
obex            650/tcp    OBEX
obex            650/udp    OBEX
ieee-mms        651/tcp    IEEE MMS
ieee-mms        651/udp    IEEE MMS
udlr-dtcp       652/tcp    UDLR_DTCP
udlr-dtcp       652/udp    UDLR_DTCP
repscmd         653/tcp    RepCmd
repscmd         653/udp    RepCmd
aodv            654/tcp    AODV
aodv            654/udp    AODV
tinc            655/tcp    TINC
tinc            655/udp    TINC
spmp            656/tcp    SPMP
spmp            656/udp    SPMP
rmc             657/tcp    RMC
rmc             657/udp    RMC
tenfold         658/tcp    TenFold
tenfold         658/udp    TenFold
url-rendezvous  659/tcp    URL Rendezvous
url-rendezvous  659/udp    URL Rendezvous
mac-srvr-admin  660/tcp    MacOS Serv Admin
mac-srvr-admin  660/udp    MacOS Ser Admin
hap             661/tcp    HAP
hap             661/udp    HAP
pftp            662/tcp    PFTP
pftp            662/udp    PFTP
purenoise       663/tcp    PureNoise
purenoise       663/udp    PureNoise
```

```
rap           256/tcp   RAP
rap           256/udp   RAP
set           257/tcp   Secure Elect Trans
set           257/udp   Secure Elect Trans
yak-chat      258/tcp   Yak Personal Chat
yak-chat      258/udp   Yak Personal Chat
esro-gen      259/tcp   Efficient Short
esro-gen      259/udp   Efficient Short
openport      260/tcp   Openport
openport      260/udp   Openport
nsiiops       261/tcp   IIOP over TLS/SSL
nsiiops       261/udp   IIOP over TLS/SSL
arcisdms      262/tcp   Arcisdms
arcisdms      262/udp   Arcisdms
hdap          263/tcp   HDAP
hdap          263/udp   HDAP
bgmp          264/tcp   BGMP
bgmp          264/udp   BGMP
x-bone-ctl    265/tcp   X-Bone CTL
x-bone-ctl    265/udp   X-Bone CTL
sst           266/tcp   SCSI on ST
sst           266/udp   SCSI on ST
td-service    267/tcp   Tobit David Layer
td-service    267/udp   Tobit David Layer
td-replica    268/tcp   Tobit David Replica
td-replica    268/udp   Tobit David Replica
#             269-279   Unassigned
http-mgmt     280/tcp   http-mgmt
http-mgmt     280/udp   http-mgmt
personal-link 281/tcp   Personal Link
personal-link 281/udp   Personal Link
cableport-ax  282/tcp   Cable Port A/X
cableport-ax  282/udp   Cable Port A/X
rescap        283/tcp   rescap
rescap        283/udp   rescap
corerjd       284/tcp   corerjd
corerjd       284/udp   corerjd
#             285       Unassigned
fxp-1         286/tcp   FXP-1
fxp-1         286/udp   FXP-1
k-block       287/tcp   K-BLOCK
k-block       287/udp   K-BLOCK
#             288-307   Unassigned
novastorbakcup  308/tcp    Novastor Backup
novastorbakcup  308/udp    Novastor Backup
entrusttime   309/tcp   EntrustTime
entrusttime   309/udp   EntrustTime
bhmds         310/tcp   bhmds
bhmds         310/udp   bhmds
asip-webadmin 311/tcp   AppleShare WebAdmin
asip-webadmin 311/udp   AppleShare WebAdmin
vslmp         312/tcp   VSLMP
vslmp         312/udp   VSLMP
magenta-logic 313/tcp   Magenta Logic
magenta-logic 313/udp   Magenta Logic
opalis-robot  314/tcp   Opalis Robot
opalis-robot  314/udp   Opalis Robot
dpsi          315/tcp   DPSI
dpsi          315/udp   DPSI
decauth       316/tcp   decAuth
decauth       316/udp   decAuth
zannet        317/tcp   Zannet
zannet        317/udp   Zannet
pkix-timestamp  318/tcp    PKIX TimeStamp
pkix-timestamp  318/udp    PKIX TimeStamp
ptp-event     319/tcp   PTP Event
ptp-event     319/udp   PTP Event
ptp-general   320/tcp   PTP General
ptp-general   320/udp   PTP General
pip           321/tcp   PIP
pip           321/udp   PIP
secure-aux-bus  664/tcp    Secure Aux Bus
secure-aux-bus  664/udp    Secure Aux Bus
sun-dr        665/tcp    Sun DR
sun-dr        665/udp    Sun DR
mdqs          666/tcp
mdqs          666/udp
doom          666/tcp    doom Id Software
doom          666/udp    doom Id Software
disclose      667/tcp    SDR Technologies
disclose      667/udp    SDR Technologies
mecomm        668/tcp    MeComm
mecomm        668/udp    MeComm
meregister    669/tcp    MeRegister
meregister    669/udp    MeRegister
vacdsm-sws    670/tcp    VACDSM-SWS
vacdsm-sws    670/udp    VACDSM-SWS
vacdsm-app    671/tcp    VACDSM-APP
vacdsm-app    671/udp    VACDSM-APP
vpps-qua      672/tcp    VPPS-QUA
vpps-qua      672/udp    VPPS-QUA
cimplex       673/tcp    CIMPLEX
cimplex       673/udp    CIMPLEX
acap          674/tcp    ACAP
acap          674/udp    ACAP
dctp          675/tcp    DCTP
dctp          675/udp    DCTP
vpps-via      676/tcp    VPPS Via
vpps-via      676/udp    VPPS Via
vpp           677/tcp    Virtual Presence
vpp           677/udp    Virtual Presence
ggf-ncp       678/tcp    GNU NCP
ggf-ncp       678/udp    GNU NCP
mrm           679/tcp    MRM
mrm           679/udp    MRM
entrust-aaas    680/tcp    entrust-aaas
entrust-aaas    680/udp    entrust-aaas
entrust-aams    681/tcp    entrust-aams
entrust-aams    681/udp    entrust-aams
xfr           682/tcp    XFR
xfr           682/udp    XFR
corba-iiop    683/tcp    CORBA IIOP
corba-iiop    683/udp    CORBA IIOP
corba-iiop-ssl  684/tcp    CORBA IIOP SSL
corba-iiop-ssl  684/udp    CORBA IIOP SSL
mdc-portmapper  685/tcp    MDC Port Mapper
mdc-portmapper  685/udp    MDC Port Mapper
hcp-wismar    686/tcp    Hardware Control
hcp-wismar    686/udp    Hardware Control
asipregistry    687/tcp    asipregistry
asipregistry    687/udp    asipregistry
realm-rusd    688/tcp    REALM-RUSD
realm-rusd    688/udp    REALM-RUSD
nmap          689/tcp    NMAP
nmap          689/udp    NMAP
vatp          690/tcp    VATP
vatp          690/udp    VATP
msexch-routing  691/tcp    MS Exchange
msexch-routing  691/udp    MS Exchange
hyperwave-isp   692/tcp    Hyperwave-ISP
hyperwave-isp   692/udp    Hyperwave-ISP
connendp      693/tcp    connendp
connendp      693/udp    connendp
ha-cluster    694/tcp    ha-cluster
ha-cluster    694/udp    ha-cluster
ieee-mms-ssl    695/tcp    IEEE-MMS-SSL
ieee-mms-ssl    695/udp    IEEE-MMS-SSL
rushd         696/tcp    RUSHD
rushd         696/udp    RUSHD
uuidgen       697/tcp    UUIDGEN
uuidgen       697/udp    UUIDGEN
olsr          698/tcp    OLSR
```

```
rtsps          322/tcp   RTSPS
rtsps          322/udp   RTSPS
#              323-332   Unassigned
texar          333/tcp   Texar Security Port
texar          333/udp   Texar Security Port
#              334-343   Unassigned
pdap           344/tcp   Prospero Data Access
pdap           344/udp   Prospero Data Access
pawserv        345/tcp   Perf Analysis Bench
pawserv        345/udp   Perf Analysis Bench
zserv          346/tcp   Zebra server
zserv          346/udp   Zebra server
fatserv        347/tcp   Fatmen Server
fatserv        347/udp   Fatmen Server
csi-sgwp       348/tcp   Cabletron Management
csi-sgwp       348/udp   Cabletron Management
mftp           349/tcp   mftp
mftp           349/udp   mftp
matip-type-a   350/tcp   MATIP Type A
matip-type-a   350/udp   MATIP Type A
matip-type-b   351/tcp   MATIP Type B
matip-type-b   351/udp   MATIP Type B
bhoetty        351/tcp   bhoetty
bhoetty        351/udp   bhoetty
dtag-ste-sb    352/tcp   DTAG
dtag-ste-sb    352/udp   DTAG
bhoedap4       352/tcp   bhoedap4
bhoedap4       352/udp   bhoedap4
ndsauth        353/tcp   NDSAUTH
ndsauth        353/udp   NDSAUTH
bh611          354/tcp   bh611
bh611          354/udp   bh611
datex-asn      355/tcp   DATEX-ASN
datex-asn      355/udp   DATEX-ASN
cloanto-net-1356/tcp   Cloanto Net 1
cloanto-net-1356/udp   Cloanto Net 1
bhevent        357/tcp   bhevent
bhevent        357/udp   bhevent
shrinkwrap     358/tcp   Shrinkwrap
shrinkwrap     358/udp   Shrinkwrap
tenebris_nts   359/tcp   Tenebris Network
tenebris_nts   359/udp   Tenebris Network
scoi2odialog   360/tcp   scoi2odialog
scoi2odialog   360/udp   scoi2odialog
semantix       361/tcp   Semantix
semantix       361/udp   Semantix
srssend        362/tcp   SRS Send
srssend        362/udp   SRS Send
rsvp_tunnel    363/tcp   RSVP Tunnel
rsvp_tunnel    363/udp   RSVP Tunnel
aurora-cmgr    364/tcp   Aurora CMGR
aurora-cmgr    364/udp   Aurora CMGR
dtk            365/tcp   DTK
dtk            365/udp   DTK
odmr           366/tcp   ODMR
odmr           366/udp   ODMR
mortgageware   367/tcp   MortgageWare
mortgageware   367/udp   MortgageWare
qbikgdp        368/tcp   QbikGDP
qbikgdp        368/udp   QbikGDP
rpc2portmap    369/tcp   rpc2portmap
rpc2portmap    369/udp   rpc2portmap
codaauth2      370/tcp   codaauth2
codaauth2      370/udp   codaauth2
clearcase      371/tcp   Clearcase
clearcase      371/udp   Clearcase
ulistproc      372/tcp   ListProcessor
ulistproc      372/udp   ListProcessor
legent-1       373/tcp   Legent Corporation
legent-1       373/udp   Legent Corporation
legent-2       374/tcp   Legent Corporation
olsr           698/udp   OLSR
#              699-703   Unassigned
elcsd          704/tcp   errlog copy
elcsd          704/udp   errlog copy
agentx         705/tcp   AgentX
agentx         705/udp   AgentX
silc           706/tcp   SILC
silc           706/udp   SILC
borland-dsj    707/tcp   Borland DSJ
borland-dsj    707/udp   Borland DSJ
#              708       Unassigned
entrust-kmsh   709/tcp   Entrust Key
entrust-kmsh   709/udp   Entrust Key
entrust-ash    710/tcp   Entrust Admin
entrust-ash    710/udp   Entrust Admin
cisco-tdp      711/tcp   Cisco TDP
cisco-tdp      711/udp   Cisco TDP
#              712-728   Unassigned
netviewdm1     729/tcp   IBM NetView serv/cli
netviewdm1     729/udp   IBM NetView serv/cli
netviewdm2     730/tcp   IBM NetView send/tcp
netviewdm2     730/udp   IBM NetView send/tcp
netviewdm3     731/tcp   IBM NetView recv/tcp
netviewdm3     731/udp   IBM NetView recv/tcp
#              732-740   Unassigned
netgw          741/tcp   netGW
netgw          741/udp   netGW
netrcs         742/tcp   Net Rev. Cont. Sys.
netrcs         742/udp   Net Rev. Cont. Sys.
#              743       Unassigned
flexlm         744/tcp   Flexible License Man
flexlm         744/udp   Flexible License Man
#              745-746   Unassigned
fujitsu-dev    747/tcp   Fujitsu Dev Ctl
fujitsu-dev    747/udp   Fujitsu Dev Ctl
ris-cm         748/tcp   Russell Info Sci
ris-cm         748/udp   Russell Info Sci
kerberos-adm   749/tcp   kerberos admin
kerberos-adm   749/udp   kerberos admin
rfile          750/tcp
loadav         750/udp
kerberos-iv    750/udp   kerberos iv
pump           751/tcp
pump           751/udp
qrh            752/tcp
qrh            752/udp
rrh            753/tcp
rrh            753/udp
tell           754/tcp   send
tell           754/udp   send
#              755-756   Unassigned
nlogin         758/tcp
nlogin         758/udp
con            759/tcp
con            759/udp
ns             760/tcp
ns             760/udp
rxe            761/tcp
rxe            761/udp
quotad         762/tcp
quotad         762/udp
cycleserv      763/tcp
cycleserv      763/udp
omserv         764/tcp
omserv         764/udp
webster        765/tcp
webster        765/udp
#              766       Unassigned
phonebook      767/tcp   phone
phonebook      767/udp   phone
#              768       Unassigned
```

```
legent-2      374/udp   Legent Corporation
hassle        375/tcp   Hassle
hassle        375/udp   Hassle
nip           376/tcp   Amiga Envoy Network
nip           376/udp   Amiga Envoy Network
tnETOS        377/tcp   NEC Corporation
tnETOS        377/udp   NEC Corporation
dsETOS        378/tcp   NEC Corporation
dsETOS        378/udp   NEC Corporation
is99c         379/tcp   TIA/EIA/IS-99 client
is99c         379/udp   TIA/EIA/IS-99 client
is99s         380/tcp   TIA/EIA/IS-99 server
is99s         380/udp   TIA/EIA/IS-99 server
hp-collector 381/tcp   hp performance data
hp-collector 381/udp   hp performance data
hp-managed-node 382/tcp    hp managed node
hp-managed-node 382/udp    hp managed node
hp-alarm-mgr    383/tcp    hp alarm manager
hp-alarm-mgr    383/udp    hp alarm manager
arns          384/tcp   Remote Net Server
arns          384/udp   Remote Net Server
ibm-app       385/tcp   IBM Application
ibm-app       385/udp   IBM Application
asa           386/tcp   ASA Message Router
asa           386/udp   ASA Message Router
aurp          387/tcp   Appletalk
aurp          387/udp   Appletalk
unidata-ldm   388/tcp   Unidata LDM
unidata-ldm   388/udp   Unidata LDM
ldap          389/tcp   LDAP
ldap          389/udp   LDAP
uis           390/tcp   UIS
uis           390/udp   UIS
synotics-relay  391/tcp    SynOptics SNMP
synotics-relay  391/udp    SynOptics SNMP
synotics-broker 392/tcp    SynOptics Port
synotics-broker 392/udp    SynOptics Port
dis           393/tcp   Data Interpretation
dis           393/udp   Data Interpretation
embl-ndt      394/tcp   EMBL Nucleic Data
embl-ndt      394/udp   MBL Nucleic Data
netcp         395/tcp   NETscout Control
netcp         395/udp   NETscout Control
netware-ip    396/tcp   Novell Netware IP
netware-ip    396/udp   Novell Netware IP
mptn          397/tcp   Multi Trans. Net.
mptn          397/udp   Multi Trans. Net.
kryptolan     398/tcp   Kryptolan
kryptolan     398/udp   Kryptolan
iso-tsap-c2   399/tcp   ISO Transport Class
iso-tsap-c2   399/udp   ISO Transport Class
work-sol      400/tcp   Workstation Sol
work-sol      400/udp   Workstation Sol
ups           401/tcp   UPS
ups           401/udp   UPS
genie         402/tcp   Genie Protocol
genie         402/udp   Genie Protocol
decap         403/tcp   decap
decap         403/udp   decap
nced          404/tcp   nced
nced          404/udp   nced
ncld          405/tcp   ncld
ncld          405/udp   ncld
imsp          406/tcp   Interactive Mail Sup
imsp          406/udp   Interactive Mail Sup
timbuktu      407/tcp   Timbuktu
timbuktu      407/udp   Timbuktu
prm-sm        408/tcp   Prospero Resource
prm-sm        408/udp   Prospero Resource
prm-nm        409/tcp   Prospero Resource
prm-nm        409/udp   Prospero Resource
vid         769/tcp
vid         769/udp
cadlock     770/tcp
cadlock     770/udp
rtip        771/tcp
rtip        771/udp
cycleserv2  772/tcp
cycleserv2  772/udp
submit      773/tcp
notify      773/udp
rpasswd     774/tcp
acmaint_dbd     774/udp
entomb      775/tcp
acmaint_transd  775/udp
wpages      776/tcp
wpages      776/udp
multiling-http  777/tcp    Multiling HTTP
multiling-http  777/udp    Multiling HTTP
#           778-779    Unassigned
wpgs        780/tcp
wpgs        780/udp
#           781-785    Unassigned
concert     786/tcp    Concert
concert     786/udp    Concert
qsc         787/tcp    QSC
qsc         787/udp    QSC
#           788-799    Unassigned
mdbs_daemon     800/tcp
mdbs_daemon     800/udp
device      801/tcp
device      801/udp
#           802-809    Unassigned
fcp-udp     810/tcp    FCP
fcp-udp     810/udp    FCP Datagram
#           811-827    Unassigned
itm-mcell-s     828/tcp    itm-mcell-s
itm-mcell-s     828/udp    itm-mcell-s
pkix-3-ca-ra    829/tcp    PKIX-3 CA/RA
pkix-3-ca-ra    829/udp    PKIX-3 CA/RA
#           830-872    Unassigned
rsync       873/tcp    rsync
rsync       873/udp    rsync
#           874-885    Unassigned
iclcnet-locate  886/tcp    ICL coNETion
iclcnet-locate  886/udp    ICL coNETion
iclcnet_svinfo  887/tcp    ICL coNETion
iclcnet_svinfo  887/udp    ICL coNETion
accessbuilder   888/tcp    AccessBuilder
accessbuilder   888/udp    AccessBuilder
cddbp           888/tcp    CD Database
#           889-899    Unassigned
omginitialrefs  900/tcp    OMG Initial Refs
omginitialrefs  900/udp    OMG Initial Refs
smpnameres      901/tcp    SMPNAMERES
smpnameres      901/udp    SMPNAMERES
ideafarm-chat   902/tcp    IDEAFARM-CHAT
ideafarm-chat   902/udp    IDEAFARM-CHAT
ideafarm-catch  903/tcp    IDEAFARM-CATCH
ideafarm-catch  903/udp    IDEAFARM-CATCH
#           904-910    Unassigned
xact-backup     911/tcp    xact-backup
xact-backup     911/udp    xact-backup
#           912-988    Unassigned
ftps-data   989/tcp    ftp protocol TLS/SSL
ftps-data   989/udp    ftp protocol TLS/SSL
ftps        990/tcp    ftp protocol TLS/SSL
ftps        990/udp    ftp protocol TLS/SSL
nas         991/tcp    Netnews Admin System
nas         991/udp    Netnews Admin System
telnets     992/tcp    telnet TLS/SSL
telnets     992/udp    telnet TLS/SSL
```

```
decladebug   410/tcp   DECLadebug
decladebug   410/udp   DECLadebug
rmt          411/tcp   Remote MT Protocol
rmt          411/udp   Remote MT Protocol
synoptics-trap  412/tcp    Trap Convention
synoptics-trap  412/udp    Trap Convention
smsp         413/tcp   SMSP
smsp         413/udp   SMSP
infoseek     414/tcp   InfoSeek
infoseek     414/udp   InfoSeek
bnet         415/tcp   BNet
bnet         415/udp   BNet
silverplatter416/tcp   Silverplatter
silverplatter416/udp   Silverplatter
onmux        417/tcp   Onmux
onmux        417/udp   Onmux
hyper-g      418/tcp   Hyper-G
hyper-g      418/udp   Hyper-G
ariel1       419/tcp   Ariel
ariel1       419/udp   Ariel
smpte        420/tcp   SMPTE
smpte        420/udp   SMPTE
ariel2       421/tcp   Ariel
ariel2       421/udp   Ariel
ariel3       422/tcp   Ariel
ariel3       422/udp   Ariel
opc-job-start423/tcp   IBM Operations
opc-job-start423/udp   IBM Operations
imaps        993/tcp    imap4 TLS/SSL
imaps        993/udp    imap4 TLS/SSL
ircs         994/tcp    irc TLS/SSL
ircs         994/udp    irc TLS/SSL
pop3s        995/tcp    pop3 TLS/SSL
pop3s        995/udp    pop3 TLS/SSL
vsinet       996/tcp    vsinet
vsinet       996/udp    vsinet
maitrd       997/tcp
maitrd       997/udp
busboy       998/tcp
puparp       998/udp
garcon       999/tcp
applix       999/udp    Applix ac
puprouter    999/tcp
puprouter    999/udp
cadlock2     1000/tcp
cadlock2     1000/udp
#            1001-1009  Unassigned
#            1008/udp   Possibly used by Sun
surf         1010/tcp   surf
surf         1010/udp   surf
#            1011-1022  Reserved
#            1023/tcp   Reserved
             1023/udp   Reserved
```

Registered / Dynamic and/or Private Ports:

Below is the list of registered as well as Dynamic and/or Private Ports. The Registered Ports are those from 1024 through 49151 and the Dynamic and/or Private Ports are those from 49152 through 65535.

Keyword	Decimal	Description
#	1024/tcp	Reserved
#	1024/udp	Reserved
blackjack	1025/tcp	network blackjack
blackjack	1025/udp	network blackjack
#	1026-1029	Unassigned
iad1	1030/tcp	BBN IAD
iad1	1030/udp	BBN IAD
iad2	1031/tcp	BBN IAD
iad2	1031/udp	BBN IAD
iad3	1032/tcp	BBN IAD
iad3	1032/udp	BBN IAD
#	1033-1046	Unassigned
neod1	1047/tcp	Sun's NEO Object
neod1	1047/udp	Sun's NEO Object
neod2	1048/tcp	Sun's NEO Object
neod2	1048/udp	Sun's NEO Object
td-postman	1049/tcp	Tobit David Postman
td-postman	1049/udp	Tobit David Postman
cma	1050/tcp	CORBA Manag Agent
cma	1050/udp	CORBA Manag Agent
optima-vnet	1051/tcp	Optima VNET
optima-vnet	1051/udp	Optima VNET
ddt	1052/tcp	Dynamic DNS Tools
ddt	1052/udp	Dynamic DNS Tools
remote-as	1053/tcp	Remote Assistant (RA)
remote-as	1053/udp	Remote Assistant (RA)
brvread	1054/tcp	BRVREAD
brvread	1054/udp	BRVREAD
ansyslmd	1055/tcp	ANSYS-License Manager
ansyslmd	1055/udp	ANSYS-License Manager
vfo	1056/tcp	VFO
vfo	1056/udp	VFO
startron	1057/tcp	STARTRON
startron	1057/udp	STARTRON
nim	1058/tcp	nim
nim	1058/udp	nim
nimreg	1059/tcp	nimreg
nimreg	1059/udp	nimreg
polestar	1060/tcp	POLESTAR
polestar	1060/udp	POLESTAR
kiosk	1061/tcp	KIOSK
kiosk	1061/udp	KIOSK
veracity	1062/tcp	Veracity
veracity	1062/udp	Veracity
kyoceranetdev	1063/tcp	KyoceraNetDev
kyoceranetdev	1063/udp	KyoceraNetDev
jstel	1064/tcp	JSTEL
jstel	1064/udp	JSTE
syscomlan	1065/tcp	SYSCOMLAN
syscomlan	1065/udp	SYSCOMLAN
fpo-fns	1066/tcp	FPO-FNS
fpo-fns	1066/udp	FPO-FNS
instl_boots	1067/tcp	Bootstrap Proto.
instl_boots	1067/udp	Bootstrap Proto.
instl_bootc	1068/tcp	Bootstrap Proto.
instl_bootc	1068/udp	Bootstrap Proto.
cognex-insight	1069/tcp	COGNEX-INSIGHT
cognex-insight	1069/udp	COGNEX-INSIGHT
gmrupdateserv	1070/tcp	GMRUpdateSERV
gmrupdateserv	1070/udp	GMRUpdateSERV
bsquare-voip	1071/tcp	BSQUARE-VOIP
bsquare-voip	1071/udp	BSQUARE-VOIP

Keyword	Decimal	Description
alarm-clock-s	2667/tcp	Alarm Clock Serv
alarm-clock-s	2667/udp	Alarm Clock Serv
alarm-clock-c	2668/tcp	Alarm Clock Clt
alarm-clock-c	2668/udp	Alarm Clock Clt
toad	2669/tcp	TOAD
toad	2669/udp	TOAD
tve-announce	2670/tcp	TVE Announce
tve-announce	2670/udp	TVE Announce
newlixreg	2671/tcp	newlixreg
newlixreg	2671/udp	newlixreg
nhserver	2672/tcp	nhserver
nhserver	2672/udp	nhserver
firstcall42	2673/tcp	First Call 42
firstcall42	2673/udp	First Call 42
ewnn	2674/tcp	ewnn
ewnn	2674/udp	ewnn
ttc-etap	2675/tcp	TTC ETAP
ttc-etap	2675/udp	TTC ETAP
simslink	2676/tcp	SIMSLink
simslink	2676/udp	SIMSLink
gadgetgate1way	2677/tcp	Gadget Gate1 Way
gadgetgate1way	2677/udp	Gadget Gate1 Way
gadgetgate2way	2678/tcp	Gadget Gate2 Way
gadgetgate2way	2678/udp	Gadget Gate2 Way
syncserverssl	2679/tcp	Sync Server SSL
syncserverssl	2679/udp	Sync Server SSL
pxc-sapxom	2680/tcp	pxc-sapxom
pxc-sapxom	2680/udp	pxc-sapxom
mpnjsomb	2681/tcp	mpnjsomb
mpnjsomb	2681/udp	mpnjsomb
srsp	2682/tcp	SRSP
srsp	2682/udp	SRSP
ncdloadbalance	2683/tcp	NCDLoadBalance
ncdloadbalance	2683/udp	NCDLoadBalance
mpnjsosv	2684/tcp	mpnjsosv
mpnjsosv	2684/udp	mpnjsosv
mpnjsocl	2685/tcp	mpnjsocl
mpnjsocl	2685/udp	mpnjsocl
mpnjsomg	2686/tcp	mpnjsomg
mpnjsomg	2686/udp	mpnjsomg
pq-lic-mgmt	2687/tcp	pq-lic-mgmt
pq-lic-mgmt	2687/udp	pq-lic-mgmt
md-cg-http	2688/tcp	md-cf-http
md-cg-http	2688/udp	md-cf-http
fastlynx	2689/tcp	FastLynx
fastlynx	2689/udp	FastLynx
hp-nnm-data	2690/tcp	HP NNM Embedded
hp-nnm-data	2690/udp	HP NNM Embedded
itinternet	2691/tcp	IT Internet
itinternet	2691/udp	IT Internet
admins-lms	2692/tcp	Admins LMS
admins-lms	2692/udp	Admins LMS
belarc-http	2693/tcp	belarc-http
belarc-http	2693/udp	belarc-http
pwrsevent	2694/tcp	pwrsevent
pwrsevent	2694/udp	pwrsevent
vspread	2695/tcp	VSPREAD
vspread	2695/udp	VSPREAD
unifyadmin	2696/tcp	Unify Admin
unifyadmin	2696/udp	Unify Admin
oce-snmp-trap	2697/tcp	Oce SNMP Trap
oce-snmp-trap	2697/udp	Oce SNMP Trap

cardax	1072/tcp	CARDAX
cardax	1072/udp	CARDA
bridgecontrol	1073/tcp	BridgeControl
bridgecontrol	1073/udp	BridgeContro
fastechnologlm	1074/tcp	FASTechnologies
fastechnologlm	1074/udp	FASTechnologie
rdrmshc	1075/tcp	RDRMSHC
rdrmshc	1075/udp	RDRMSHC
dab-sti-c	1076/tcp	DAB STI-C
dab-sti-c	1076/udp	DAB STI-C
imgames	1077/tcp	IMGames
imgames	1077/udp	IMGames
emanagecstp	1078/tcp	eManageCstp
emanagecstp	1078/udp	eManageCst
asprovatalk	1079/tcp	ASPROVATalk
asprovatalk	1079/udp	ASPROVATalk
socks	1080/tcp	Socks
socks	1080/udp	Socks
amt-esd-prot	1082/tcp	AMT-ESD-PROT
amt-esd-prot	1082/udp	AMT-ESD-PROT
ansoft-lm-1	1083/tcp	Anasoft
ansoft-lm-1	1083/udp	Anasoft
ansoft-lm-2	1084/tcp	Anasoft
ansoft-lm-2	1084/udp	Anasoft
webobjects	1085/tcp	Web Objects
webobjects	1085/udp	Web Objects
cplscrambler-lg	1086/tcp	CPL Scramble
cplscrambler-lg	1086/udp	CPL Scrambler
cplscrambler-in	1087/tcp	CPL Scrambler
cplscrambler-in	1087/udp	CPL Scramble
cplscrambler-al	1088/tcp	CPL Scrambler
cplscrambler-al	1088/udp	CPL Scramble
ff-annunc	1089/tcp	FF Annunciation
ff-annunc	1089/udp	FF Annunciation
ff-fms	1090/tcp	FF Fieldbus
ff-fms	1090/udp	FF Fieldbus
ff-sm	1091/tcp	FF System Manag
ff-sm	1091/udp	FF System Manag
obrpd	1092/tcp	OBRPD
obrpd	1092/udp	OBRPD
proofd	1093/tcp	PROOFD
proofd	1093/udp	PROOFD
rootd	1094/tcp	ROOTD
rootd	1094/udp	ROOTD
nicelink	1095/tcp	NICELink
nicelink	1095/udp	NICELink
cnrprotocol	1096/tcp	Common Name Resl
cnrprotocol	1096/udp	Common Name Resl
sunclustermgr	1097/tcp	Sun Cluster Man
sunclustermgr	1097/udp	Sun Cluster Man
rmiactivation	1098/tcp	RMI Activation
rmiactivation	1098/udp	RMI Activation
rmiregistry	1099/tcp	RMI Registry
rmiregistry	1099/udp	RMI Registry
mctp	1100/tcp	MCTP
mctp	1100/udp	MCTP
pt2-discover	1101/tcp	PT2-DISCOVER
pt2-discover	1101/udp	PT2-DISCOVER
adobeserver-1	1102/tcp	ADOBE SERVER 1
adobeserver-1	1102/udp	ADOBE SERVER 1
adobeserver-2	1103/tcp	ADOBE SERVER 2
adobeserver-2	1103/udp	ADOBE SERVER 2
xrl	1104/tcp	XRL
xrl	1104/udp	XRL
ftranhc	1105/tcp	FTRANHC
ftranhc	1105/udp	FTRANHC
isoipsigport-1	1106/tcp	ISOIPSIGPORT-1
isoipsigport-1	1106/udp	ISOIPSIGPORT-1
isoipsigport-2	1107/tcp	ISOIPSIGPORT-2
isoipsigport-2	1107/udp	ISOIPSIGPORT-2
ratio-adp	1108/tcp	ratio-adp
mck-ivpip	2698/tcp	MCK-IVPIP
mck-ivpip	2698/udp	MCK-IVPIP
csoft-plusclnt	2699/tcp	Csoft Plus Clt
csoft-plusclnt	2699/udp	Csoft Plus Clt
tqdata	2700/tcp	tqdata
tqdata	2700/udp	tqdata
sms-rcinfo	2701/tcp	SMS RCINFO
sms-rcinfo	2701/udp	SMS RCINFO
sms-xfer	2702/tcp	SMS XFER
sms-xfer	2702/udp	SMS XFER
sms-chat	2703/tcp	SMS CHAT
sms-chat	2703/udp	SMS CHAT
sms-remctrl	2704/tcp	SMS REMCTRL
sms-remctrl	2704/udp	SMS REMCTRL
sds-admin	2705/tcp	SDS Admin
sds-admin	2705/udp	SDS Admin
ncdmirroring	2706/tcp	NCD Mirroring
ncdmirroring	2706/udp	NCD Mirroring
emcsymapiport	2707/tcp	EMCSYMAPIPORT
emcsymapiport	2707/udp	EMCSYMAPIPORT
banyan-net	2708/tcp	Banyan-Net
banyan-net	2708/udp	Banyan-Net
supermon	2709/tcp	Supermon
supermon	2709/udp	Supermon
sso-service	2710/tcp	SSO Service
sso-service	2710/udp	SSO Service
sso-control	2711/tcp	SSO Control
sso-control	2711/udp	SSO Control
aocp	2712/tcp	Axapta Object
aocp	2712/udp	Axapta Object
raven1	2713/tcp	Raven1
raven1	2713/udp	Raven1
raven2	2714/tcp	Raven2
raven2	2714/udp	Raven2
hpstgmgr2	2715/tcp	HPSTGMGR2
hpstgmgr2	2715/udp	HPSTGMGR2
inova-ip-disco	2716/tcp	Inova IP Disco
inova-ip-disco	2716/udp	Inova IP Disco
pn-requester	2717/tcp	PN REQUESTER
pn-requester	2717/udp	PN REQUESTER
pn-requester2	2718/tcp	PN REQUESTER 2
pn-requester2	2718/udp	PN REQUESTER 2
scan-change	2719/tcp	Scan & Change
scan-change	2719/udp	Scan & Change
wkars	2720/tcp	wkars
wkars	2720/udp	wkars
smart-diagnose	2721/tcp	Smart Diagnose
smart-diagnose	2721/udp	Smart Diagnose
proactivesrvr	2722/tcp	Proactive Server
proactivesrvr	2722/udp	Proactive Server
watchdognt	2723/tcp	WatchDog NT
watchdognt	2723/udp	WatchDog NT
qotps	2724/tcp	qotps
qotps	2724/udp	qotps
msolap-ptp2	2725/tcp	MSOLAP PTP2
msolap-ptp2	2725/udp	MSOLAP PTP2
tams	2726/tcp	TAMS
tams	2726/udp	TAMS
mgcp-callagent	2727/tcp	Media Gateway
mgcp-callagent	2727/udp	Media Gateway
sqdr	2728/tcp	SQDR
sqdr	2728/udp	SQDR
tcim-control	2729/tcp	TCIM Control
tcim-control	2729/udp	TCIM Control
nec-raidplus	2730/tcp	NEC RaidPlus
nec-raidplus	2730/udp	NEC RaidPlus
netdragon-msngr	2731/tcp	NetDragon Mes
netdragon-msngr	2731/udp	NetDragon Mes
g5m	2732/tcp	G5M
g5m	2732/udp	G5M
signet-ctf	2733/tcp	Signet CTF

ratio-adp	1108/udp	ratio-adp
#	1109	Unassigned
nfsd-status	1110/tcp	Cluster status
nfsd-keepalive	1110/udp	Client status
lmsocialserver	1111/tcp	LM Social Server
lmsocialserver	1111/udp	LM Social Server
icp	1112/tcp	Intelligent Com
icp	1112/udp	Intelligent Com
#	1113	Unassigned
mini-sql	1114/tcp	Mini SQL
mini-sql	1114/udp	Mini SQL
ardus-trns	1115/tcp	ARDUS Transfer
ardus-trns	1115/udp	ARDUS Transfer
ardus-cntl	1116/tcp	ARDUS Control
ardus-cntl	1116/udp	ARDUS Control
ardus-mtrns	1117/tcp	ARDUS Multicast
ardus-mtrns	1117/udp	ARDUS Multicast
#	1118-1122	Unassigned
murray	1123/tcp	Murray
murray	1123/udp	Murray
#	1124-1154	Unassigned
nfa	1155/tcp	Network File Acs
nfa	1155/udp	Network File Acs
#	1156-1160	Unassigned
health-polling	1161/tcp	Health Polling
health-polling	1161/udp	Health Polling
health-trap	1162/tcp	Health Trap
health-trap	1162/udp	Health Trap
#	1163-1168	Unassigned
tripwire	1169/tcp	TRIPWIRE
tripwire	1169/udp	TRIPWIRE
#	1170-1179	Unassigned
mc-client	1180/tcp	Millicent Proxy
mc-client	1180/udp	Millicent Proxy
#	1181-1187	Unassigned
hp-webadmin	1188/tcp	HP Web Admin
hp-webadmin	1188/udp	HP Web Admin
#	1189-1199	Unassigned
scol	1200/tcp	SCOL
scol	1200/udp	SCOL
nucleus-sand	1201/tcp	Nucleus Sand
nucleus-sand	1201/udp	Nucleus Sand
caiccipc	1202/tcp	caiccipc
caiccipc	1202/udp	caiccipc
ssslic-mgr	1203/tcp	License Valid
ssslic-mgr	1203/udp	License Valid
ssslog-mgr	1204/tcp	Log Request
ssslog-mgr	1204/udp	Log Request
accord-mgc	1205/tcp	Accord-MGC
accord-mgc	1205/udp	Accord-MGC
anthony-data	1206/tcp	Anthony Data
anthony-data	1206/udp	Anthony Data
metasage	1207/tcp	MetaSage
metasage	1207/udp	MetaSage
seagull-ais	1208/tcp	SEAGULL AIS
seagull-ais	1208/udp	SEAGULL AIS
ipcd3	1209/tcp	IPCD3
ipcd3	1209/udp	IPCD3
eoss	1210/tcp	EOSS
eoss	1210/udp	EOSS
groove-dpp	1211/tcp	Groove DPP
groove-dpp	1211/udp	Groove DPP
lupa	1212/tcp	lupa
lupa	1212/udp	lupa
mpc-lifenet	1213/tcp	MPC LIFENET
mpc-lifenet	1213/udp	MPC LIFENET
kazaa	1214/tcp	KAZAA
kazaa	1214/udp	KAZAA
scanstat-1	1215/tcp	scanSTAT 1.0
scanstat-1	1215/udp	scanSTAT 1.0
etebac5	1216/tcp	ETEBAC 5
signet-ctf	2733/udp	Signet CTF
ccs-software	2734/tcp	CCS Software
ccs-software	2734/udp	CCS Software
monitorconsole	2735/tcp	Monitor Console
monitorconsole	2735/udp	Monitor Console
radwiz-nms-srv	2736/tcp	RADWIZ NMS SRV
radwiz-nms-srv	2736/udp	RADWIZ NMS SRV
srp-feedback	2737/tcp	SRP Feedback
srp-feedback	2737/udp	SRP Feedback
ndl-tcp-ois-gw	2738/tcp	NDL TCP-OSI Gty
ndl-tcp-ois-gw	2738/udp	NDL TCP-OSI Gty
tn-timing	2739/tcp	TN Timing
tn-timing	2739/udp	TN Timing
alarm	2740/tcp	Alarm
alarm	2740/udp	Alarm
tsb	2741/tcp	TSB
tsb	2741/udp	TSB
tsb2	2742/tcp	TSB2
tsb2	2742/udp	TSB2
murx	2743/tcp	murx
murx	2743/udp	murx
honyaku	2744/tcp	honyaku
honyaku	2744/udp	honyaku
urbisnet	2745/tcp	URBISNET
urbisnet	2745/udp	URBISNET
cpudpencap	2746/tcp	CPUDPENCAP
cpudpencap	2746/udp	CPUDPENCAP
fjippol-swrly	2747/tcp	
fjippol-swrly	2747/udp	
fjippol-polsvr	2748/tcp	
fjippol-polsvr	2748/udp	
fjippol-cnsl	2749/tcp	
fjippol-cnsl	2749/udp	
fjippol-port1	2750/tcp	
fjippol-port1	2750/udp	
fjippol-port2	2751/tcp	
fjippol-port2	2751/udp	
rsisysaccess	2752/tcp	RSISYS ACCESS
rsisysaccess	2752/udp	RSISYS ACCESS
de-spot	2753/tcp	de-spot
de-spot	2753/udp	de-spot
apollo-cc	2754/tcp	APOLLO CC
apollo-cc	2754/udp	APOLLO CC
expresspay	2755/tcp	Express Pay
expresspay	2755/udp	Express Pay
simplement-tie	2756/tcp	simplement-tie
simplement-tie	2756/udp	simplement-tie
cnrp	2757/tcp	CNRP
cnrp	2757/udp	CNRP
apollo-status	2758/tcp	APOLLO Status
apollo-status	2758/udp	APOLLO Status
apollo-gms	2759/tcp	APOLLO GMS
apollo-gms	2759/udp	APOLLO GMS
sabams	2760/tcp	Saba MS
sabams	2760/udp	Saba MS
dicom-iscl	2761/tcp	DICOM ISCL
dicom-iscl	2761/udp	DICOM ISCL
dicom-tls	2762/tcp	DICOM TLS
dicom-tls	2762/udp	DICOM TLS
desktop-dna	2763/tcp	Desktop DNA
desktop-dna	2763/udp	Desktop DNA
data-insurance	2764/tcp	Data Insurance
data-insurance	2764/udp	Data Insurance
qip-audup	2765/tcp	qip-audup
qip-audup	2765/udp	qip-audup
compaq-scp	2766/tcp	Compaq SCP
compaq-scp	2766/udp	Compaq SCP
uadtc	2767/tcp	UADTC
uadtc	2767/udp	UADTC
uacs	2768/tcp	UACS
uacs	2768/udp	UACS

etebac5	1216/udp	ETEBAC 5
hpss-ndapi	1217/tcp	HPSS-NDAPI
hpss-ndapi	1217/udp	HPSS-NDAPI
aeroflight-ads	1218/tcp	AeroFlight-ADs
aeroflight-ads	1218/udp	AeroFlight-ADs
aeroflight-ret	1219/tcp	AeroFlight-Ret
aeroflight-ret	1219/udp	AeroFlight-Ret
qt-serveradmin	1220/tcp	QT SERVER ADMIN
qt-serveradmin	1220/udp	QT SERVER ADMIN
sweetware-apps	1221/tcp	SweetWARE Apps
sweetware-apps	1221/udp	SweetWARE Apps
nerv	1222/tcp	SNI R&D network
nerv	1222/udp	SNI R&D network
tgp	1223/tcp	TGP
tgp	1223/udp	TGP
vpnz	1224/tcp	VPNz
vpnz	1224/udp	VPNz
slinkysearch	1225/tcp	SLINKYSEARCH
slinkysearch	1225/udp	SLINKYSEARCH
stgxfws	1226/tcp	STGXFWS
stgxfws	1226/udp	STGXFWS
dns2go	1227/tcp	DNS2Go
dns2go	1227/udp	DNS2Go
florence	1228/tcp	FLORENCE
florence	1228/udp	FLORENCE
novell-zfs	1229/tcp	Novell ZFS
novell-zfs	1229/udp	Novell ZFS
periscope	1230/tcp	Periscope
periscope	1230/udp	Periscope
menandmice-lpm	1231/tcp	menandmice-lpm
menandmice-lpm	1231/udp	menandmice-lpm
mtrgtrans	1232/tcp	mtrgtrans
mtrgtrans	1232/udp	mtrgtrans
univ-appserver	1233/tcp	Universal App
univ-appserver	1233/udp	Universal App
search-agent	1234/tcp	Infoseek Search
search-agent	1234/udp	Infoseek Search
#	1235-1238	Unassigned
nmsd	1239/tcp	NMSD
nmsd	1239/udp	NMSD
#	1240-1247	Unassigned
hermes	1248/tcp	
hermes	1248/udp	
#	1249-1299	Unassigned
h323hostcallsc	1300/tcp	H323 Host Call
h323hostcallsc	1300/udp	H323 Host Call
#	1301-1309	Unassigned
husky	1310/tcp	Husky
husky	1310/udp	Husky
rxmon	1311/tcp	RxMon
rxmon	1311/udp	RxMon
sti-envision	1312/tcp	STI Envision
sti-envision	1312/udp	STI Envision
bmc_patroldb	1313/tcp	BMC_PATROLDB
bmc-patroldb	1313/udp	BMC_PATROLDB
pdps	1314/tcp	Photoscript
pdps	1314/udp	Photoscript
#	1315-1318	Unassigned
panja-icsp	1319/tcp	Panja-ICSP
panja-icsp	1319/udp	Panja-ICSP
panja-axbnet	1320/tcp	Panja-AXBNET
panja-axbnet	1320/udp	Panja-AXBNET
pip	1321/tcp	PIP
pip	1321/udp	PIP
#	1322-1334	Unassigned
digital-notary	1335/tcp	Digital Notary
digital-notary	1335/udp	Digital Notary
#	1336-1344	Unassigned
vpjp	1345/tcp	VPJP
vpjp	1345/udp	VPJP
alta-ana-lm	1346/tcp	Alta Analytics
singlept-mvs	2769/tcp	Single Point MVS
singlept-mvs	2769/udp	Single Point MV
veronica	2770/tcp	Veronica
veronica	2770/udp	Veronica
vergencecm	2771/tcp	Vergence CM
vergencecm	2771/udp	Vergence C
auris	2772/tcp	auris
auris	2772/udp	auris
pcbakcup1	2773/tcp	PC Backup
pcbakcup1	2773/udp	PC Backup
pcbakcup2	2774/tcp	PC Backup
pcbakcup2	2774/udp	PC Backup
smpp	2775/tcp	SMMP
smpp	2775/udp	SMMP
ridgeway1	2776/tcp	Ridgeway
ridgeway1	2776/udp	Ridgeway
ridgeway2	2777/tcp	Ridgeway
ridgeway2	2777/udp	Ridgeway
gwen-sonya	2778/tcp	Gwen-Sonya
gwen-sonya	2778/udp	Gwen-Sonya
lbc-sync	2779/tcp	LBC Sync
lbc-sync	2779/udp	LBC Sync
lbc-control	2780/tcp	LBC Control
lbc-control	2780/udp	LBC Control
whosells	2781/tcp	whosells
whosells	2781/udp	whosells
everydayrc	2782/tcp	everydayrc
everydayrc	2782/udp	everydayrc
aises	2783/tcp	AISES
aises	2783/udp	AISES
www-dev	2784/tcp	world wide web
www-dev	2784/udp	world wide web
aic-np	2785/tcp	aic-np
aic-np	2785/udp	aic-np
aic-oncrpc	2786/tcp	aic-oncrpc
aic-oncrpc	2786/udp	aic-oncrpc
piccolo	2787/tcp	piccolo
piccolo	2787/udp	piccolo
fryeserv	2788/tcp	NetWare Loadable
fryeserv	2788/udp	NetWare Loadable
media-agent	2789/tcp	Media Agent
media-agent	2789/udp	Media Agent
plgproxy	2790/tcp	PLG Proxy
plgproxy	2790/udp	PLG Proxy
mtport-regist	2791/tcp	MT Port Regist
mtport-regist	2791/udp	MT Port Regist
f5-globalsite	2792/tcp	f5-globalsite
f5-globalsite	2792/udp	f5-globalsite
initlsmsad	2793/tcp	initlsmsad
initlsmsad	2793/udp	initlsmsad
aaftp	2794/tcp	aaftp
aaftp	2794/udp	aaftp
livestats	2795/tcp	LiveStats
livestats	2795/udp	LiveStats
ac-tech	2796/tcp	ac-tech
ac-tech	2796/udp	ac-tech
esp-encap	2797/tcp	esp-encap
esp-encap	2797/udp	esp-encap
tmesis-upshot	2798/tcp	TMESIS-UPShot
tmesis-upshot	2798/udp	TMESIS-UPShot
icon-discover	2799/tcp	ICON Discover
icon-discover	2799/udp	ICON Discover
acc-raid	2800/tcp	ACC RAID
acc-raid	2800/udp	ACC RAID
igcp	2801/tcp	IGCP
igcp	2801/udp	IGCP
veritas-tcp1	2802/tcp	Veritas TCP1
veritas-udp1	2802/udp	Veritas UDP1
btprjctrl	2803/tcp	btprjctrl
btprjctrl	2803/udp	btprjctrl
telexis-vtu	2804/tcp	Telexis VTU

alta-ana-lm	1346/udp	Alta Analytics
bbn-mmc	1347/tcp	multi media conf
bbn-mmc	1347/udp	multi media conf
bbn-mmx	1348/tcp	multi media conf
bbn-mmx	1348/udp	multi media conf
sbook	1349/tcp	Registration Net
sbook	1349/udp	Registration Net
editbench	1350/tcp	Registration Net
editbench	1350/udp	Registration Net
equationbuilder	1351/tcp	Digital Works
equationbuilder	1351/udp	Digital Works
lotusnote	1352/tcp	Lotus Note
lotusnote	1352/udp	Lotus Note
relief	1353/tcp	Relief Consult
relief	1353/udp	Relief Consult
rightbrain	1354/tcp	RightBrain Soft
rightbrain	1354/udp	RightBrain Soft
intuitive-edge	1355/tcp	Intuitive Edge
intuitive-edge	1355/udp	Intuitive Edge
cuillamartin	1356/tcp	CuillaMartin
cuillamartin	1356/udp	CuillaMartin
pegboard	1357/tcp	Elect PegBoard
pegboard	1357/udp	Elect PegBoard
connlcli	1358/tcp	CONNLCLI
connlcli	1358/udp	CONNLCLI
ftsrv	1359/tcp	FTSRV
ftsrv	1359/udp	FTSRV
mimer	1360/tcp	MIMER
mimer	1360/udp	MIMER
linx	1361/tcp	LinX
linx	1361/udp	LinX
timeflies	1362/tcp	TimeFlies
timeflies	1362/udp	TimeFlies
ndm-requester	1363/tcp	DataMover Req
ndm-requester	1363/udp	DataMover Req
ndm-server	1364/tcp	DataMover Server
ndm-server	1364/udp	DataMover Server
adapt-sna	1365/tcp	Software Ass
adapt-sna	1365/udp	Software Ass
netware-csp	1366/tcp	Novell NetWare
netware-csp	1366/udp	Novell NetWare
dcs	1367/tcp	DCS
dcs	1367/udp	DCS
screencast	1368/tcp	ScreenCast
screencast	1368/udp	ScreenCast
gv-us	1369/tcp	GV to Unix Shell
gv-us	1369/udp	GV to Unix Shell
us-gv	1370/tcp	Unix Shell to GV
us-gv	1370/udp	Unix Shell to GV
fc-cli	1371/tcp	Fujitsu Config
fc-cli	1371/udp	Fujitsu Config
fc-ser	1372/tcp	Fujitsu Config
fc-ser	1372/udp	Fujitsu Config
chromagrafx	1373/tcp	Chromagrafx
chromagrafx	1373/udp	Chromagrafx
molly	1374/tcp	EPI Software Sys
molly	1374/udp	EPI Software Sys
bytex	1375/tcp	Bytex
bytex	1375/udp	Bytex
ibm-pps	1376/tcp	IBM Pers to Pers
ibm-pps	1376/udp	IBM Pers to Pers
cichlid	1377/tcp	Cichlid
cichlid	1377/udp	Cichlid
elan	1378/tcp	Elan
elan	1378/udp	Elan
dbreporter	1379/tcp	Integrity Sol
dbreporter	1379/udp	Integrity Sol
telesis-licman	1380/tcp	Telesis Network
telesis-licman	1380/udp	Telesis Network
apple-licman	1381/tcp	Apple Network
apple-licman	1381/udp	Apple Network
telexis-vtu	2804/udp	Telexis VTU
wta-wsp-s	2805/tcp	WTA WSP-S
wta-wsp-s	2805/udp	WTA WSP-S
cspuni	2806/tcp	cspuni
cspuni	2806/udp	cspuni
cspmulti	2807/tcp	cspmulti
cspmulti	2807/udp	cspmulti
j-lan-p	2808/tcp	J-LAN-P
j-lan-p	2808/udp	J-LAN-P
corbaloc	2809/tcp	CORBA LOC
corbaloc	2809/udp	CORBA LOC
netsteward	2810/tcp	Active Net
netsteward	2810/udp	Active Net
gsiftp	2811/tcp	GSI FTP
gsiftp	2811/udp	GSI FTP
atmtcp	2812/tcp	atmtcp
atmtcp	2812/udp	atmtcp
llm-pass	2813/tcp	llm-pass
llm-pass	2813/udp	llm-pass
llm-csv	2814/tcp	llm-csv
llm-csv	2814/udp	llm-csv
lbc-measure	2815/tcp	LBC Measurement
lbc-measure	2815/udp	LBC Measurement
lbc-watchdog	2816/tcp	LBC Watchdog
lbc-watchdog	2816/udp	LBC Watchdog
nmsigport	2817/tcp	NMSig Port
nmsigport	2817/udp	NMSig Port
rmlnk	2818/tcp	rmlnk
rmlnk	2818/udp	rmlnk
fc-faultnotify	2819/tcp	FC Fault Notif
fc-faultnotify	2819/udp	FC Fault Notif
univision	2820/tcp	UniVision
univision	2820/udp	UniVision
vml-dms	2821/tcp	vml_dms
vml-dms	2821/udp	vml_dms
ka0wuc	2822/tcp	ka0wuc
ka0wuc	2822/udp	ka0wuc
cqg-netlan	2823/tcp	CQG Net/LAN
cqg-netlan	2823/udp	CQG Net/LAN
slc-systemlog	2826/tcp	slc systemlog
slc-systemlog	2826/udp	slc systemlog
slc-ctrlrloops	2827/tcp	slc ctrlrloops
slc-ctrlrloops	2827/udp	slc ctrlrloops
itm-lm	2828/tcp	ITM License Mgr
itm-lm	2828/udp	ITM License Mgr
silkp1	2829/tcp	silkp1
silkp1	2829/udp	silkp1
silkp2	2830/tcp	silkp2
silkp2	2830/udp	silkp2
silkp3	2831/tcp	silkp3
silkp3	2831/udp	silkp3
silkp4	2832/tcp	silkp4
silkp4	2832/udp	silkp4
glishd	2833/tcp	glishd
glishd	2833/udp	glishd
evtp	2834/tcp	EVTP
evtp	2834/udp	EVTP
evtp-data	2835/tcp	EVTP-DATA
evtp-data	2835/udp	EVTP-DATA
catalyst	2836/tcp	catalyst
catalyst	2836/udp	catalyst
repliweb	2837/tcp	Repliweb
repliweb	2837/udp	Repliweb
starbot	2838/tcp	Starbot
starbot	2838/udp	Starbot
nmsigport	2839/tcp	NMSigPort
nmsigport	2839/udp	NMSigPort
l3-exprt	2840/tcp	l3-exprt
l3-exprt	2840/udp	l3-exprt
l3-ranger	2841/tcp	l3-ranger
l3-ranger	2841/udp	l3-ranger

udt_os	1382/tcp	
udt_os	1382/udp	
gwha	1383/tcp	GW Hannaway
gwha	1383/udp	GW Hannaway
os-licman	1384/tcp	Objective Sol
os-licman	1384/udp	Objective Sol
atex_elmd	1385/tcp	Atex Publishing
atex_elmd	1385/udp	Atex Publishing
checksum	1386/tcp	CheckSum
checksum	1386/udp	CheckSum
cadsi-lm	1387/tcp	Computer Aided
cadsi-lm	1387/udp	Computer Aided
objective-dbc	1388/tcp	Objective Sol
objective-dbc	1388/udp	Objective Sol
iclpv-dm	1389/tcp	Document Manager
iclpv-dm	1389/udp	Document Manager
iclpv-sc	1390/tcp	Storage Ctl
iclpv-sc	1390/udp	Storage Ctl
iclpv-sas	1391/tcp	Storage Access
iclpv-sas	1391/udp	Storage Access
iclpv-pm	1392/tcp	Print Manager
iclpv-pm	1392/udp	Print Manager
iclpv-nls	1393/tcp	Network Log Serv
iclpv-nls	1393/udp	Network Log Serv
iclpv-nlc	1394/tcp	Network Log Clt
iclpv-nlc	1394/udp	Network Log Clt
iclpv-wsm	1395/tcp	PC Workstation
iclpv-wsm	1395/udp	PC Workstation
dvl-activemail	1396/tcp	DVL Active Mail
dvl-activemail	1396/udp	DVL Active Mail
audio-activmail	1397/tcp	Audio Act Mail
audio-activmail	1397/udp	Audio Act Mail
video-activmail	1398/tcp	Video Act Mail
video-activmail	1398/udp	Video Act Mail
cadkey-licman	1399/tcp	Cadkey
cadkey-licman	1399/udp	Cadkey
cadkey-tablet	1400/tcp	Cadkey
cadkey-tablet	1400/udp	Cadkey
goldleaf-licman	1401/tcp	Goldleaf
goldleaf-licman	1401/udp	Goldleaf
prm-sm-np	1402/tcp	Prospero Res Man
prm-sm-np	1402/udp	Prospero Res Man
prm-nm-np	1403/tcp	Prospero Res Man
prm-nm-np	1403/udp	Prospero Res Man
igi-lm	1404/tcp	Infinite Graph
igi-lm	1404/udp	Infinite Graph
ibm-res	1405/tcp	IBM Remote Exec
ibm-res	1405/udp	IBM Remote Exec
netlabs-lm	1406/tcp	NetLabs
netlabs-lm	1406/udp	NetLabs
dbsa-lm	1407/tcp	DBSA
dbsa-lm	1407/udp	DBSA
sophia-lm	1408/tcp	Sophia
sophia-lm	1408/udp	Sophia
here-lm	1409/tcp	Here License Man
here-lm	1409/udp	Here License Man
hiq	1410/tcp	HiQ License Man
hiq	1410/udp	HiQ License Mana
af	1411/tcp	AudioFile
af	1411/udp	AudioFile
innosys	1412/tcp	InnoSys
innosys	1412/udp	InnoSys
innosys-acl	1413/tcp	Innosys-ACL
innosys-acl	1413/udp	Innosys-ACL
ibm-mqseries	1414/tcp	IBM MQSeries
ibm-mqseries	1414/udp	IBM MQSeries
dbstar	1415/tcp	DBStar
dbstar	1415/udp	DBStar
novell-lu6.2	1416/tcp	Novell LU6.2
novell-lu6.2	1416/udp	Novell LU6.2
timbuktu-srv1	1417/tcp	Timbuktu Serv 1
l3-hawk	2842/tcp	l3-hawk
l3-hawk	2842/udp	l3-hawk
pdnet	2843/tcp	PDnet
pdnet	2843/udp	PDnet
bpcp-poll	2844/tcp	BPCP POLL
bpcp-poll	2844/udp	BPCP POLL
bpcp-trap	2845/tcp	BPCP TRAP
bpcp-trap	2845/udp	BPCP TRAP
aimpp-hello	2846/tcp	AIMPP Hello
aimpp-hello	2846/udp	AIMPP Hello
aimpp-port-req	2847/tcp	AIMPP Port Req
aimpp-port-req	2847/udp	AIMPP Port Req
amt-blc-port	2848/tcp	AMT-BLC-PORT
amt-blc-port	2848/udp	AMT-BLC-PORT
fxp	2849/tcp	FXP
fxp	2849/udp	FXP
metaconsole	2850/tcp	MetaConsole
metaconsole	2850/udp	MetaConsole
webemshttp	2851/tcp	webemshttp
webemshttp	2851/udp	webemshttp
bears-01	2852/tcp	bears-01
bears-01	2852/udp	bears-01
ispipes	2853/tcp	ISPipes
ispipes	2853/udp	ISPipes
infomover	2854/tcp	InfoMover
infomover	2854/udp	InfoMover
cesdinv	2856/tcp	cesdinv
cesdinv	2856/udp	cesdinv
simctlp	2857/tcp	SimCtIP
simctlp	2857/udp	SimCtIP
ecnp	2858/tcp	ECNP
ecnp	2858/udp	ECNP
activememory	2859/tcp	Active Memory
activememory	2859/udp	Active Memory
dialpad-voice1	2860/tcp	Dialpad Voice 1
dialpad-voice1	2860/udp	Dialpad Voice 1
dialpad-voice2	2861/tcp	Dialpad Voice 2
dialpad-voice2	2861/udp	Dialpad Voice 2
ttg-protocol	2862/tcp	TTG Protocol
ttg-protocol	2862/udp	TTG Protocol
sonardata	2863/tcp	Sonar Data
sonardata	2863/udp	Sonar Data
astromed-main	2864/tcp	main 5001 cmd
astromed-main	2864/udp	main 5001 cmd
pit-vpn	2865/tcp	pit-vpn
pit-vpn	2865/udp	pit-vpn
lwlistener	2866/tcp	lwlistener
lwlistener	2866/udp	lwlistener
esps-portal	2867/tcp	esps-portal
esps-portal	2867/udp	esps-portal
npep-messaging	2868/tcp	NPEP Messaging
npep-messaging	2868/udp	NPEP Messaging
icslap	2869/tcp	ICSLAP
icslap	2869/udp	ICSLAP
daishi	2870/tcp	daishi
daishi	2870/udp	daishi
msi-selectplay	2871/tcp	MSI Select Play
msi-selectplay	2871/udp	MSI Select Play
contract	2872/tcp	CONTRACT
contract	2872/udp	CONTRACT
paspar2-zoomin	2873/tcp	PASPAR2 ZoomIn
paspar2-zoomin	2873/udp	PASPAR2 ZoomIn
dxmessagebase1	2874/tcp	dxmessagebase1
dxmessagebase1	2874/udp	dxmessagebase1
dxmessagebase2	2875/tcp	dxmessagebase2
dxmessagebase2	2875/udp	dxmessagebase2
sps-tunnel	2876/tcp	SPS Tunnel
sps-tunnel	2876/udp	SPS Tunnel
bluelance	2877/tcp	BLUELANCE
bluelance	2877/udp	BLUELANCE
aap	2878/tcp	AAP

timbuktu-srv1	1417/udp	Timbuktu Serv 1
timbuktu-srv2	1418/tcp	Timbuktu Serv 2
timbuktu-srv2	1418/udp	Timbuktu Serv 2
timbuktu-srv3	1419/tcp	Timbuktu Serv 3
timbuktu-srv3	1419/udp	Timbuktu Serv 3
timbuktu-srv4	1420/tcp	Timbuktu Serv 4
timbuktu-srv4	1420/udp	Timbuktu Serv 4
gandalf-lm	1421/tcp	Gandalf
gandalf-lm	1421/udp	Gandalf
autodesk-lm	1422/tcp	Autodesk
autodesk-lm	1422/udp	Autodesk
essbase	1423/tcp	Essbase Arbor
essbase	1423/udp	Essbase Arbor
hybrid	1424/tcp	Hybrid Encrypt
hybrid	1424/udp	Hybrid Encrypt
zion-lm	1425/tcp	Zion Software
zion-lm	1425/udp	Zion Software
sais	1426/tcp	Satellite-data 1
sais	1426/udp	Satellite-data 1
mloadd	1427/tcp	mloadd
mloadd	1427/udp	mloadd
informatik-lm	1428/tcp	Informatik
informatik-lm	1428/udp	Informatik
nms	1429/tcp	Hypercom NMS
nms	1429/udp	Hypercom NMS
tpdu	1430/tcp	Hypercom TPDU
tpdu	1430/udp	Hypercom TPDU
rgtp	1431/tcp	Reverse Gossip
rgtp	1431/udp	Reverse Gossip
blueberry-lm	1432/tcp	Blueberry Soft
blueberry-lm	1432/udp	Blueberry Soft
ms-sql-s	1433/tcp	Microsoft-SQL
ms-sql-s	1433/udp	Microsoft-SQL
ms-sql-m	1434/tcp	Microsoft-SQL
ms-sql-m	1434/udp	Microsoft-SQL
ibm-cics	1435/tcp	IBM CICS
ibm-cics	1435/udp	IBM CICS
saism	1436/tcp	Satellite-data 2
saism	1436/udp	Satellite-data 2
tabula	1437/tcp	Tabula
tabula	1437/udp	Tabul
eicon-server	1438/tcp	Eicon Security
eicon-server	1438/udp	Eicon Security
eicon-x25	1439/tcp	Eicon X25/SNA
eicon-x25	1439/udp	Eicon X25/SNA
eicon-slp	1440/tcp	Eicon Service
eicon-slp	1440/udp	Eicon Service
cadis-1	1441/tcp	Cadis
cadis-1	1441/udp	Cadis
cadis-2	1442/tcp	Cadis
cadis-2	1442/udp	Cadis
ies-lm	1443/tcp	Int Eng Soft
ies-lm	1443/udp	Int Eng Soft
marcam-lm	1444/tcp	Marcam
marcam-lm	1444/udp	Marcam
proxima-lm	1445/tcp	Proxima
proxima-lm	1445/udp	Proxima
ora-lm	1446/tcp	Optical Research
ora-lm	1446/udp	Optical Research
apri-lm	1447/tcp	Applied Parallel
apri-lm	1447/udp	Applied Parallel
oc-lm	1448/tcp	OpenConnect
oc-lm	1448/udp	OpenConnect
peport	1449/tcp	PEport
peport	1449/udp	PEport
dwf	1450/tcp	Tandem
dwf	1450/udp	Tandem
infoman	1451/tcp	IBM Information
infoman	1451/udp	IBM Information
gtegsc-lm	1452/tcp	GTE Government
gtegsc-lm	1452/udp	GTE Government
aap	2878/udp	AAP
ucentric-ds	2879/tcp	ucentric-ds
ucentric-ds	2879/udp	ucentric-ds
synapse	2880/tcp	synapse
synapse	2880/udp	synapse
ndsp	2881/tcp	NDSP
ndsp	2881/udp	NDSP
ndtp	2882/tcp	NDTP
ndtp	2882/udp	NDTP
ndnp	2883/tcp	NDNP
ndnp	2883/udp	NDNP
flashmsg	2884/tcp	Flash Msg
flashmsg	2884/udp	Flash Msg
topflow	2885/tcp	TopFlow
topflow	2885/udp	TopFlow
responselogic	2886/tcp	RESPONSELOGIC
responselogic	2886/udp	RESPONSELOGIC
aironetddp	2887/tcp	aironet
aironetddp	2887/udp	aironet
spcsdlobby	2888/tcp	SPCSDLOBBY
spcsdlobby	2888/udp	SPCSDLOBBY
rsom	2889/tcp	RSOM
rsom	2889/udp	RSOM
cspclmulti	2890/tcp	CSPCLMULTI
cspclmulti	2890/udp	CSPCLMULTI
cinegrfx-elmd	2891/tcp	CINEGRFX-ELMD
cinegrfx-elmd	2891/udp	CINEGRFX-ELMD
snifferdata	2892/tcp	SNIFFERDATA
snifferdata	2892/udp	SNIFFERDATA
vseconnector	2893/tcp	VSECONNECTOR
vseconnector	2893/udp	VSECONNECTOR
abacus-remote	2894/tcp	ABACUS-REMOTE
abacus-remote	2894/udp	ABACUS-REMOTE
natuslink	2895/tcp	NATUS LINK
natuslink	2895/udp	NATUS LINK
ecovisiong6-1	2896/tcp	ECOVISIONG6-1
ecovisiong6-1	2896/udp	ECOVISIONG6-1
citrix-rtmp	2897/tcp	Citrix RTMP
citrix-rtmp	2897/udp	Citrix RTMP
appliance-cfg	2898/tcp	APPLIANCE-CFG
appliance-cfg	2898/udp	APPLIANCE-CFG
powergemplus	2899/tcp	POWERGEMPLUS
powergemplus	2899/udp	POWERGEMPLUS
quicksuite	2900/tcp	QUICKSUITE
quicksuite	2900/udp	QUICKSUITE
allstorcns	2901/tcp	ALLSTORCNS
allstorcns	2901/udp	ALLSTORCNS
netaspi	2902/tcp	NET ASPI
netaspi	2902/udp	NET ASPI
suitcase	2903/tcp	SUITCASE
suitcase	2903/udp	SUITCASE
m2ua	2904/tcp	M2UA
m2ua	2904/udp	M2UA
m3ua	2905/tcp	M3UA
m3ua	2905/udp	M3UA
caller9	2906/tcp	CALLER9
caller9	2906/udp	CALLER9
webmethods-b2b	2907/tcp	WEBMETHODS B2B
webmethods-b2b	2907/udp	WEBMETHODS B2B
mao	2908/tcp	mao
mao	2908/udp	mao
funk-dialout	2909/tcp	Funk Dialout
funk-dialout	2909/udp	Funk Dialout
tdaccess	2910/tcp	TDAccess
tdaccess	2910/udp	TDAccess
blockade	2911/tcp	Blockade
blockade	2911/udp	Blockade
epicon	2912/tcp	Epicon
epicon	2912/udp	Epicon
boosterware	2913/tcp	Booster Ware
boosterware	2913/udp	Booster Ware

genie-lm	1453/tcp	Genie
genie-lm	1453/udp	Genie
interhdl_elmd	1454/tcp	interHDL
interhdl_elmd	1454/udp	interHDL
esl-lm	1455/tcp	ESL
esl-lm	1455/udp	ESL
dca	1456/tcp	DCA
dca	1456/udp	DCA
valisys-lm	1457/tcp	Valisys
valisys-lm	1457/udp	Valisys
nrcabq-lm	1458/tcp	Nichols Research
nrcabq-lm	1458/udp	Nichols Research
proshare1	1459/tcp	Proshare App
proshare1	1459/udp	Proshare App
proshare2	1460/tcp	Proshare App
proshare2	1460/udp	Proshare App
ibm_wrless_lan	1461/tcp	IBM Wireless LAN
ibm_wrless_lan	1461/udp	IBM Wireless LAN
world-lm	1462/tcp	World
world-lm	1462/udp	World
nucleus	1463/tcp	Nucleus
nucleus	1463/udp	Nucleus
msl_lmd	1464/tcp	MSL License Man
msl_lmd	1464/udp	MSL License Man
pipes	1465/tcp	Pipes Platform
pipes	1465/udp	Pipes Platform
oceansoft-lm	1466/tcp	Ocean Software
oceansoft-lm	1466/udp	Ocean Software
csdmbase	1467/tcp	CSDMBASE
csdmbase	1467/udp	CSDMBASE
csdm	1468/tcp	CSDM
csdm	1468/udp	CSDM
aal-lm	1469/tcp	Active Analysis
aal-lm	1469/udp	Active Analysis
uaiact	1470/tcp	Univ Analytics
uaiact	1470/udp	Univ Analytics
csdmbase	1471/tcp	csdmbase
csdmbase	1471/udp	csdmbase
csdm	1472/tcp	csdm
csdm	1472/udp	csdm
openmath	1473/tcp	OpenMath
openmath	1473/udp	OpenMath
telefinder	1474/tcp	Telefinder
telefinder	1474/udp	Telefinder
taligent-lm	1475/tcp	Taligent
taligent-lm	1475/udp	Taligent
clvm-cfg	1476/tcp	clvm-cfg
clvm-cfg	1476/udp	clvm-cfg
ms-sna-server	1477/tcp	ms-sna-server
ms-sna-server	1477/udp	ms-sna-server
ms-sna-base	1478/tcp	ms-sna-base
ms-sna-base	1478/udp	ms-sna-base
dberegister	1479/tcp	dberegister
dberegister	1479/udp	dberegister
pacerforum	1480/tcp	PacerForum
pacerforum	1480/udp	PacerForum
airs	1481/tcp	AIRS
airs	1481/udp	AIRS
miteksys-lm	1482/tcp	Miteksys
miteksys-lm	1482/udp	Miteksys
afs	1483/tcp	AFS
afs	1483/udp	AFS
confluent	1484/tcp	Confluent
confluent	1484/udp	Confluent
lansource	1485/tcp	LANSource
lansource	1485/udp	LANSource
nms_topo_serv	1486/tcp	nms_topo_serv
nms_topo_serv	1486/udp	nms_topo_serv
localinfosrvr	1487/tcp	LocalInfoSrvr
localinfosrvr	1487/udp	LocalInfoSrvr
docstor	1488/tcp	DocStor
gamelobby	2914/tcp	Game Lobby
gamelobby	2914/udp	Game Lobby
tksocket	2915/tcp	TK Socket
tksocket	2915/udp	TK Socket
elvin_server	2916/tcp	Elvin Server
elvin_server	2916/udp	Elvin Server
elvin_client	2917/tcp	Elvin Client
elvin_client	2917/udp	Elvin Client
kastenchasepad	2918/tcp	Kasten Chase Pad
kastenchasepad	2918/udp	Kasten Chase Pad
roboer	2919/tcp	ROBOER
roboer	2919/udp	ROBOER
roboeda	2920/tcp	ROBOEDA
roboeda	2920/udp	ROBOEDA
cesdcdman	2921/tcp	CESD Contents
cesdcdman	2921/udp	CESD Contents
cesdcdtrn	2922/tcp	CESD Contents
cesdcdtrn	2922/udp	CESD Contents
wta-wsp-wtp-s	2923/tcp	WTA-WSP-WTP-S
wta-wsp-wtp-s	2923/udp	WTA-WSP-WTP-S
precise-vip	2924/tcp	PRECISE-VIP
precise-vip	2924/udp	PRECISE-VIP
frp	2925/tcp	Firewall Redund
frp	2925/udp	Firewall Redund
mobile-file-dl	2926/tcp	MOBILE-FILE-DL
mobile-file-dl	2926/udp	MOBILE-FILE-DL
unimobilectrl	2927/tcp	UNIMOBILECTRL
unimobilectrl	2927/udp	UNIMOBILECTRL
redstone-cpss	2928/tcp	REDSTONE-CPSS
redstone-cpss	2928/udp	REDSONTE-CPSS
panja-webadmin	2929/tcp	PANJA-WEBADMIN
panja-webadmin	2929/udp	PANJA-WEBADMIN
panja-weblinx	2930/tcp	PANJA-WEBLINX
panja-weblinx	2930/udp	PANJA-WEBLINX
circle-x	2931/tcp	Circle-X
circle-x	2931/udp	Circle-X
incp	2932/tcp	INCP
incp	2932/udp	INCP
4-tieropmgw	2933/tcp	4-TIER OPM GW
4-tieropmgw	2933/udp	4-TIER OPM GW
4-tieropmcli	2934/tcp	4-TIER OPM CLI
4-tieropmcli	2934/udp	4-TIER OPM CLI
qtp	2935/tcp	QTP
qtp	2935/udp	QTP
otpatch	2936/tcp	OTPatch
otpatch	2936/udp	OTPatch
pnaconsult-lm	2937/tcp	PNACONSULT-LM
pnaconsult-lm	2937/udp	PNACONSULT-LM
sm-pas-1	2938/tcp	SM-PAS-1
sm-pas-1	2938/udp	SM-PAS-1
sm-pas-2	2939/tcp	SM-PAS-2
sm-pas-2	2939/udp	SM-PAS-2
sm-pas-3	2940/tcp	SM-PAS-3
sm-pas-3	2940/udp	SM-PAS-3
sm-pas-4	2941/tcp	SM-PAS-4
sm-pas-4	2941/udp	SM-PAS-4
sm-pas-5	2942/tcp	SM-PAS-5
sm-pas-5	2942/udp	SM-PAS-5
ttnrepository	2943/tcp	TTNRepository
ttnrepository	2943/udp	TTNRepository
megaco-h248	2944/tcp	Megaco H-248
megaco-h248	2944/udp	Megaco H-248
h248-binary	2945/tcp	H248 Binary
h248-binary	2945/udp	H248 Binary
fjsvmpor	2946/tcp	FJSVmpor
fjsvmpor	2946/udp	FJSVmpor
gpsd	2947/tcp	GPSD
gpsd	2947/udp	GPSD
wap-push	2948/tcp	WAP PUSH
wap-push	2948/udp	WAP PUSH
wap-pushsecure	2949/tcp	WAP PUSH SECURE

docstor	1488/udp	DocStor
dmdocbroker	1489/tcp	dmdocbroker
dmdocbroker	1489/udp	dmdocbroker
insitu-conf	1490/tcp	insitu-conf
insitu-conf	1490/udp	insitu-conf
anynetgateway	1491/tcp	anynetgateway
anynetgateway	1491/udp	anynetgateway
stone-design-1	1492/tcp	stone-design-1
stone-design-1	1492/udp	stone-design-1
netmap_lm	1493/tcp	netmap_lm
netmap_lm	1493/udp	netmap_lm
ica	1494/tcp	ica
ica	1494/udp	ica
cvc	1495/tcp	cvc
cvc	1495/udp	cvc
liberty-lm	1496/tcp	liberty-lm
liberty-lm	1496/udp	liberty-lm
rfx-lm	1497/tcp	rfx-lm
rfx-lm	1497/udp	rfx-lm
sybase-sqlany	1498/tcp	Sybase SQL Any
sybase-sqlany	1498/udp	Sybase SQL Any
fhc	1499/tcp	Federico Heinz
fhc	1499/udp	Federico Heinz
vlsi-lm	1500/tcp	VLSI
vlsi-lm	1500/udp	VLSI
saiscm	1501/tcp	Satellite-data 3
saiscm	1501/udp	Satellite-data 3
shivadiscovery	1502/tcp	Shiva
shivadiscovery	1502/udp	Shiva
imtc-mcs	1503/tcp	Databeam
imtc-mcs	1503/udp	Databeam
evb-elm	1504/tcp	EVB Software
evb-elm	1504/udp	EVB Software
funkproxy	1505/tcp	Funk Software
funkproxy	1505/udp	Funk Software
utcd	1506/tcp	Universal Time
utcd	1506/udp	Universal Time
symplex	1507/tcp	Symplex
symplex	1507/udp	Symplex
diagmond	1508/tcp	diagmond
diagmond	1508/udp	diagmond
robcad-lm	1509/tcp	Robcad, Ltd.
robcad-lm	1509/udp	Robcad, Ltd.
mvx-lm	1510/tcp	Midland Valley
mvx-lm	1510/udp	Midland Valley
3l-l1	1511/tcp	3l-l1
3l-l1	1511/udp	3l-l1
wins	1512/tcp	Name Service
wins	1512/udp	Name Service
fujitsu-dtc	1513/tcp	Fujitsu Systems
fujitsu-dtc	1513/udp	Fujitsu Systems
fujitsu-dtcns	1514/tcp	Fujitsu Systems
fujitsu-dtcns	1514/udp	Fujitsu Systems
ifor-protocol	1515/tcp	ifor-protocol
ifor-protocol	1515/udp	ifor-protocol
vpad	1516/tcp	Virtual Places
vpad	1516/udp	Virtual Places
vpac	1517/tcp	Virtual Places
vpac	1517/udp	Virtual Places
vpvd	1518/tcp	Virtual Places
vpvd	1518/udp	Virtual Places
vpvc	1519/tcp	Virtual Places
vpvc	1519/udp	Virtual Places
atm-zip-office	1520/tcp	atm zip office
atm-zip-office	1520/udp	atm zip office
ncube-lm	1521/tcp	nCube
ncube-lm	1521/udp	nCube
ricardo-lm	1522/tcp	Ricardo North
ricardo-lm	1522/udp	Ricardo North
cichild-lm	1523/tcp	cichild
cichild-lm	1523/udp	cichild
wap-pushsecure	2949/udp	WAP PUSH SECURE
esip	2950/tcp	ESIP
esip	2950/udp	ESIP
ottp	2951/tcp	OTTP
ottp	2951/udp	OTTP
mpfwsas	2952/tcp	MPFWSAS
mpfwsas	2952/udp	MPFWSAS
ovalarmsrv	2953/tcp	OVALARMSRV
ovalarmsrv	2953/udp	OVALARMSRV
ovalarmsrv-cmd	2954/tcp	OVALARMSRV-CMD
ovalarmsrv-cmd	2954/udp	OVALARMSRV-CMD
csnotify	2955/tcp	CSNOTIFY
csnotify	2955/udp	CSNOTIFY
ovrimosdbman	2956/tcp	OVRIMOSDBMAN
ovrimosdbman	2956/udp	OVRIMOSDBMAN
jmact5	2957/tcp	JAMCT5
jmact5	2957/udp	JAMCT5
jmact6	2958/tcp	JAMCT6
jmact6	2958/udp	JAMCT6
rmopagt	2959/tcp	RMOPAGT
rmopagt	2959/udp	RMOPAGT
dfoxserver	2960/tcp	DFOXSERVER
dfoxserver	2960/udp	DFOXSERVER
boldsoft-lm	2961/tcp	BOLDSOFT-LM
boldsoft-lm	2961/udp	BOLDSOFT-LM
iph-policy-cli	2962/tcp	IPH-POLICY-CLI
iph-policy-cli	2962/udp	IPH-POLICY-CLI
iph-policy-adm	2963/tcp	IPH-POLICY-ADM
iph-policy-adm	2963/udp	IPH-POLICY-ADM
bullant-srap	2964/tcp	BULLANT SRAP
bullant-srap	2964/udp	BULLANT SRAP
bullant-rap	2965/tcp	BULLANT RAP
bullant-rap	2965/udp	BULLANT RAP
idp-infotrieve	2966/tcp	IDP-INFOTRIEVE
idp-infotrieve	2966/udp	IDP-INFOTRIEVE
ssc-agent	2967/tcp	SSC-AGENT
ssc-agent	2967/udp	SSC-AGENT
enpp	2968/tcp	ENPP
enpp	2968/udp	ENPP
essp	2969/tcp	ESSP
essp	2969/udp	ESSP
index-net	2970/tcp	INDEX-NET
index-net	2970/udp	INDEX-NET
netclip	2971/tcp	Net Clip
netclip	2971/udp	Net Clip
pmsm-webrctl	2972/tcp	PMSM Webrctl
pmsm-webrctl	2972/udp	PMSM Webrctl
svnetworks	2973/tcp	SV Networks
svnetworks	2973/udp	SV Networks
signal	2974/tcp	Signal
signal	2974/udp	Signal
fjmpcm	2975/tcp	Fujitsu
fjmpcm	2975/udp	Fujitsu
cns-srv-port	2976/tcp	CNS Server Port
cns-srv-port	2976/udp	CNS Server Port
ttc-etap-ns	2977/tcp	TTCs Enterprise
ttc-etap-ns	2977/udp	TTCs Enterprise
ttc-etap-ds	2978/tcp	TTCs Enterprise
ttc-etap-ds	2978/udp	TTCs Enterprise
h263-video	2979/tcp	H.263 Video
h263-video	2979/udp	H.263 Video
wimd	2980/tcp	Instant
wimd	2980/udp	Instant
mylxamport	2981/tcp	MYLXAMPORT
mylxamport	2981/udp	MYLXAMPORT
iwb-whiteboard	2982/tcp	IWB-WHITEBOARD
iwb-whiteboard	2982/udp	IWB-WHITEBOARD
netplan	2983/tcp	NETPLAN
netplan	2983/udp	NETPLAN
hpidsadmin	2984/tcp	HPIDSADMIN
hpidsadmin	2984/udp	HPIDSADMIN

ingreslock	1524/tcp	ingres
ingreslock	1524/udp	ingres
orasrv	1525/tcp	oracle
orasrv	1525/udp	oracle
prospero-np	1525/tcp	Prospero
prospero-np	1525/udp	Prospero
pdap-np	1526/tcp	Prospero
pdap-np	1526/udp	Prospero
tlisrv	1527/tcp	oracle
tlisrv	1527/udp	oracle
mciautoreg	1528/tcp	micautoreg
mciautoreg	1528/udp	micautoreg
coauthor	1529/tcp	oracle
coauthor	1529/udp	oracle
rap-service	1530/tcp	rap-service
rap-service	1530/udp	rap-service
rap-listen	1531/tcp	rap-listen
rap-listen	1531/udp	rap-listen
miroconnect	1532/tcp	miroconnect
miroconnect	1532/udp	miroconnect
virtual-places	1533/tcp	Virtual Places
virtual-places	1533/udp	Virtual Places
micromuse-lm	1534/tcp	micromuse-lm
micromuse-lm	1534/udp	micromuse-lm
ampr-info	1535/tcp	ampr-info
ampr-info	1535/udp	ampr-info
ampr-inter	1536/tcp	ampr-inter
ampr-inter	1536/udp	ampr-inter
sdsc-lm	1537/tcp	isi-lm
sdsc-lm	1537/udp	isi-lm
3ds-lm	1538/tcp	3ds-lm
3ds-lm	1538/udp	3ds-lm
intellistor-lm	1539/tcp	Intellistor
intellistor-lm	1539/udp	Intellistor
rds	1540/tcp	rds
rds	1540/udp	rds
rds2	1541/tcp	rds2
rds2	1541/udp	rds2
gridgen-elmd	1542/tcp	gridgen-elmd
gridgen-elmd	1542/udp	gridgen-elmd
simba-cs	1543/tcp	simba-cs
simba-cs	1543/udp	simba-cs
aspeclmd	1544/tcp	aspeclmd
aspeclmd	1544/udp	aspeclmd
vistium-share	1545/tcp	vistium-share
vistium-share	1545/udp	vistium-share
abbaccuray	1546/tcp	abbaccuray
abbaccuray	1546/udp	abbaccuray
laplink	1547/tcp	laplink
laplink	1547/udp	laplink
axon-lm	1548/tcp	Axon
axon-lm	1548/udp	Axon
shivahose	1549/tcp	Shiva Hose
shivasound	1549/udp	Shiva Sound
3m-image-lm	1550/tcp	Image 3M
3m-image-lm	1550/udp	Image 3M
hecmtl-db	1551/tcp	HECMTL-DB
hecmtl-db	1551/udp	HECMTL-DB
pciarray	1552/tcp	pciarray
pciarray	1552/udp	pciarray
sna-cs	1553/tcp	sna-cs
sna-cs	1553/udp	sna-cs
caci-lm	1554/tcp	CACI Products
caci-lm	1554/udp	CACI Products
livelan	1555/tcp	livelan
livelan	1555/udp	livelan
ashwin	1556/tcp	AshWin CI
ashwin	1556/udp	AshWin CI
arbortext-lm	1557/tcp	ArborText
arbortext-lm	1557/udp	ArborText
xingmpeg	1558/tcp	xingmpeg
hpidsagent	2985/tcp	HPIDSAGENT
hpidsagnet	2985/udp	HPIDSAGENT
stonefalls	2986/tcp	STONEFALLS
stonefalls	2986/udp	STONEFALLS
identify	2987/tcp	IDENTIFY
identify	2987/udp	IDENTIFY
classify	2988/tcp	CLASSIFY
classify	2988/udp	CLASSIFY
zarkov	2989/tcp	ZARKOV
zarkov	2989/udp	ZARKOV
boscap	2990/tcp	BOSCAP
boscap	2990/udp	BOSCAP
wkstn-mon	2991/tcp	WKSTN-MON
wkstn-mon	2991/udp	WKSTN-MON
itb301	2992/tcp	ITB301
itb301	2992/udp	ITB301
veritas-vis1	2993/tcp	VERITAS VIS1
veritas-vis1	2993/udp	VERITAS VIS1
veritas-vis2	2994/tcp	VERITAS VIS2
veritas-vis2	2994/udp	VERITAS VIS2
idrs	2995/tcp	IDRS
idrs	2995/udp	IDRS
vsixml	2996/tcp	vsixml
vsixml	2996/udp	vsixml
rebol	2997/tcp	REBOL
rebol	2997/udp	REBOL
realsecure	2998/tcp	Real Secure
realsecure	2998/udp	Real Secure
remoteware-un	2999/tcp	RemoteWare
remoteware-un	2999/udp	RemoteWare
hbci	3000/tcp	HBCI
hbci	3000/udp	HBCI
remoteware-cl	3000/tcp	RemoteWare Clt
remoteware-cl	3000/udp	RemoteWare Clt
redwood-broker	3001/tcp	Redwood Broker
redwood-broker	3001/udp	Redwood Broker
exlm-agent	3002/tcp	EXLM Agent
exlm-agent	3002/udp	EXLM Agent
remoteware-srv	3002/tcp	RemoteWare Serv
remoteware-srv	3002/udp	RemoteWare Serv
cgms	3003/tcp	CGMS
cgms	3003/udp	CGMS
csoftragent	3004/tcp	Csoft Agent
csoftragent	3004/udp	Csoft Agent
geniuslm	3005/tcp	Genius
geniuslm	3005/udp	Genius
ii-admin	3006/tcp	Instant Internet
ii-admin	3006/udp	Instant Internet
lotusmtap	3007/tcp	Lotus Mail
lotusmtap	3007/udp	Lotus Mail
midnight-tech	3008/tcp	Midnight Tech
midnight-tech	3008/udp	Midnight Techn
pxc-ntfy	3009/tcp	PXC-NTFY
pxc-ntfy	3009/udp	PXC-NTFY
gw	3010/tcp	Telerate Workst
ping-pong	3010/udp	Telerate Workst
trusted-web	3011/tcp	Trusted Web
trusted-web	3011/udp	Trusted Web
twsdss	3012/tcp	Trusted Web Clt
twsdss	3012/udp	Trusted Web Clt
gilatskysurfer	3013/tcp	Gilat Sky Surfer
gilatskysurfer	3013/udp	Gilat Sky Surfer
broker_service	3014/tcp	Broker Service
broker_service	3014/udp	Broker Service
nati-dstp	3015/tcp	NATI DSTP
nati-dstp	3015/udp	NATI DSTP
notify_srvr	3016/tcp	Notify Server
notify_srvr	3016/udp	Notify Server
event_listener	3017/tcp	Event Listener
event_listener	3017/udp	Event Listener
srvc_registry	3018/tcp	Service Registry

xingmpeg	1558/udp	xingmpeg
web2host	1559/tcp	web2host
web2host	1559/udp	web2host
asci-val	1560/tcp	asci-val
asci-val	1560/udp	asci-val
facilityview	1561/tcp	facilityview
facilityview	1561/udp	facilityview
pconnectmgr	1562/tcp	pconnectmgr
pconnectmgr	1562/udp	pconnectmgr
cadabra-lm	1563/tcp	Cadabra
cadabra-lm	1563/udp	Cadabra
pay-per-view	1564/tcp	Pay-Per-View
pay-per-view	1564/udp	Pay-Per-View
winddlb	1565/tcp	WinDD
winddlb	1565/udp	WinDD
corelvideo	1566/tcp	CORELVIDEO
corelvideo	1566/udp	CORELVIDEO
jlicelmd	1567/tcp	jlicelmd
jlicelmd	1567/udp	jlicelmd
tsspmap	1568/tcp	tsspmap
tsspmap	1568/udp	tsspmap
ets	1569/tcp	ets
ets	1569/udp	ets
orbixd	1570/tcp	orbixd
orbixd	1570/udp	orbixd
rdb-dbs-disp	1571/tcp	Oracle Rem DB
rdb-dbs-disp	1571/udp	Oracle Rem DB
chip-lm	1572/tcp	Chipcom License
chip-lm	1572/udp	Chipcom License
itscomm-ns	1573/tcp	itscomm-ns
itscomm-ns	1573/udp	itscomm-ns
mvel-lm	1574/tcp	mvel-lm
mvel-lm	1574/udp	mvel-lm
oraclenames	1575/tcp	oraclenames
oraclenames	1575/udp	oraclenames
moldflow-lm	1576/tcp	moldflow-lm
moldflow-lm	1576/udp	moldflow-lm
hypercube-lm	1577/tcp	hypercube-lm
hypercube-lm	1577/udp	hypercube-lm
jacobus-lm	1578/tcp	Jacobus
jacobus-lm	1578/udp	Jacobus
ioc-sea-lm	1579/tcp	ioc-sea-lm
ioc-sea-lm	1579/udp	ioc-sea-lm
tn-tl-r1	1580/tcp	tn-tl-r1
tn-tl-r2	1580/udp	tn-tl-r2
mil-2045-47001	1581/tcp	MIL-2045-47001
mil-2045-47001	1581/udp	MIL-2045-47001
msims	1582/tcp	MSIMS
msims	1582/udp	MSIMS
simbaexpress	1583/tcp	simbaexpress
simbaexpress	1583/udp	simbaexpress
tn-tl-fd2	1584/tcp	tn-tl-fd2
tn-tl-fd2	1584/udp	tn-tl-fd2
intv	1585/tcp	intv
intv	1585/udp	intv
ibm-abtact	1586/tcp	ibm-abtact
ibm-abtact	1586/udp	ibm-abtact
pra_elmd	1587/tcp	pra_elmd
pra_elmd	1587/udp	pra_elmd
triquest-lm	1588/tcp	triquest-lm
triquest-lm	1588/udp	triquest-lm
vqp	1589/tcp	VQP
vqp	1589/udp	VQPMcCloghrie
gemini-lm	1590/tcp	gemini-lm
gemini-lm	1590/udp	gemini-lm
ncpm-pm	1591/tcp	ncpm-pm
ncpm-pm	1591/udp	ncpm-pm
commonspace	1592/tcp	commonspace
commonspace	1592/udp	commonspace
mainsoft-lm	1593/tcp	mainsoft-lm
mainsoft-lm	1593/udp	mainsoft-lm
srvc_registry	3018/udp	Service Registry
resource_mgr	3019/tcp	Resource Manager
resource_mgr	3019/udp	Resource Manager
cifs	3020/tcp	CIFS
cifs	3020/udp	CIFS
agriserver	3021/tcp	AGRI Server
agriserver	3021/udp	AGRI Server
csregagent	3022/tcp	CSREGAGENT
csregagent	3022/udp	CSREGAGENT
magicnotes	3023/tcp	magicnotes
magicnotes	3023/udp	magicnotes
nds_sso	3024/tcp	NDS_SSO
nds_sso	3024/udp	NDS_SSO
arepa-raft	3025/tcp	Arepa Raft
arepa-raft	3025/udp	Arepa Raft
agri-gateway	3026/tcp	AGRI Gateway
agri-gateway	3026/udp	AGRI Gateway
LiebDevMgmt_C	3027/tcp	LiebDevMgmt_C
LiebDevMgmt_C	3027/udp	LiebDevMgmt_C
LiebDevMgmt_DM	3028/tcp	LiebDevMgmt_DM
LiebDevMgmt_DM	3028/udp	LiebDevMgmt_DM
LiebDevMgmt_A	3029/tcp	LiebDevMgmt_A
LiebDevMgmt_A	3029/udp	LiebDevMgmt_A
arepa-cas	3030/tcp	Arepa Cas
arepa-cas	3030/udp	Arepa Cas
agentvu	3031/tcp	AgentVU
agentvu	3031/udp	AgentVU
redwood-chat	3032/tcp	Redwood Chat
redwood-chat	3032/udp	Redwood Chat
pdb	3033/tcp	PDB
pdb	3033/udp	PDB
osmosis-aeea	3034/tcp	Osmosis AEEA
osmosis-aeea	3034/udp	Osmosis AEEA
fjsv-gssagt	3035/tcp	FJSV gssagt
fjsv-gssagt	3035/udp	FJSV gssagt
hagel-dump	3036/tcp	Hagel DUMP
hagel-dump	3036/udp	Hagel DUMP
hp-san-mgmt	3037/tcp	HP SAN Mgmt
hp-san-mgmt	3037/udp	HP SAN Mgmt
santak-ups	3038/tcp	Santak UPS
santak-ups	3038/udp	Santak UPS
cogitate	3039/tcp	Cogitate, Inc.
cogitate	3039/udp	Cogitate, Inc.
tomato-springs	3040/tcp	Tomato Springs
tomato-springs	3040/udp	Tomato Springs
di-traceware	3041/tcp	di-traceware
di-traceware	3041/udp	di-traceware
journee	3042/tcp	journee
journee	3042/udp	journee
brp	3043/tcp	BRP
brp	3043/udp	BRP
responsenet	3045/tcp	ResponseNet
responsenet	3045/udp	ResponseNet
di-ase	3046/tcp	di-ase
di-ase	3046/udp	di-ase
hlserver	3047/tcp	Fast Security HL
hlserver	3047/udp	Fast Security HL
pctrader	3048/tcp	Sierra Net PC
pctrader	3048/udp	Sierra Net PC
nsws	3049/tcp	NSWS
nsws	3049/udp	NSWS
gds_db	3050/tcp	gds_db
gds_db	3050/udp	gds_db
galaxy-server	3051/tcp	Galaxy Server
galaxy-server	3051/udp	Galaxy Server
apcpcns	3052/tcp	APCPCNS
apcpcns	3052/udp	APCPCNS
dsom-server	3053/tcp	dsom-server
dsom-server	3053/udp	dsom-server
amt-cnf-prot	3054/tcp	AMT CNF PROT
amt-cnf-prot	3054/udp	AMT CNF PROT

sixtrak	1594/tcp	sixtrak
sixtrak	1594/udp	sixtrak
radio	1595/tcp	radio
radio	1595/udp	radio
radio-sm	1596/tcp	radio-sm
radio-bc	1596/udp	radio-bc
orbplus-iiop	1597/tcp	orbplus-iiop
orbplus-iiop	1597/udp	orbplus-iiop
picknfs	1598/tcp	picknfs
picknfs	1598/udp	picknfs
simbaservices	1599/tcp	simbaservices
simbaservices	1599/udp	simbaservices
issd	1600/tcp	
issd	1600/udp	
aas	1601/tcp	aas
aas	1601/udp	aas
inspect	1602/tcp	inspect
inspect	1602/udp	inspect
picodbc	1603/tcp	pickodbc
picodbc	1603/udp	pickodbc
icabrowser	1604/tcp	icabrowser
icabrowser	1604/udp	icabrowser
slp	1605/tcp	Salutation
slp	1605/udp	Salutation
slm-api	1606/tcp	Salutation
slm-api	1606/udp	Salutation
stt	1607/tcp	stt
stt	1607/udp	stt
smart-lm	1608/tcp	Smart Corp.
smart-lm	1608/udp	Smart Corp.
isysg-lm	1609/tcp	isysg-lm
isysg-lm	1609/udp	isysg-lm
taurus-wh	1610/tcp	taurus-wh
taurus-wh	1610/udp	taurus-wh
ill	1611/tcp	Inter Library
ill	1611/udp	Inter Library
netbill-trans	1612/tcp	NetBill
netbill-trans	1612/udp	NetBill
netbill-keyrep	1613/tcp	NetBill Key
netbill-keyrep	1613/udp	NetBill Key
netbill-cred	1614/tcp	NetBill
netbill-cred	1614/udp	NetBill
netbill-auth	1615/tcp	NetBill
netbill-auth	1615/udp	NetBill
netbill-prod	1616/tcp	NetBill
netbill-prod	1616/udp	NetBill
nimrod-agent	1617/tcp	Nimrod
nimrod-agent	1617/udp	Nimrod
skytelnet	1618/tcp	skytelnet
skytelnet	1618/udp	skytelnet
xs-openstorage	1619/tcp	xs-openstorage
xs-openstorage	1619/udp	xs-openstorage
faxportwinport	1620/tcp	faxportwinport
faxportwinport	1620/udp	faxportwinport
softdataphone	1621/tcp	softdataphone
softdataphone	1621/udp	softdataphone
ontime	1622/tcp	ontime
ontime	1622/udp	ontime
jaleosnd	1623/tcp	jaleosnd
jaleosnd	1623/udp	jaleosnd
udp-sr-port	1624/tcp	udp-sr-port
udp-sr-port	1624/udp	udp-sr-port
svs-omagent	1625/tcp	svs-omagent
svs-omagent	1625/udp	svs-omagent
shockwave	1626/tcp	Shockwave
shockwave	1626/udp	Shockwave
t128-gateway	1627/tcp	T.128 Gateway
t128-gateway	1627/udp	T.128 Gateway
lontalk-norm	1628/tcp	LonTalk normal
lontalk-norm	1628/udp	LonTalk normal
lontalk-urgnt	1629/tcp	LonTalk urgent
policyserver	3055/tcp	Policy Server
policyserver	3055/udp	Policy Server
cdl-server	3056/tcp	CDL Server
cdl-server	3056/udp	CDL Server
goahead-fldup	3057/tcp	GoAhead FldUp
goahead-fldup	3057/udp	GoAhead FldUp
videobeans	3058/tcp	videobeans
videobeans	3058/udp	videobeans
qsoft	3059/tcp	qsoft
qsoft	3059/tcp	qsoft
interserver	3060/tcp	interserver
interserver	3060/udp	interserver
cautcpd	3061/tcp	cautcpd
cautcpd	3061/udp	cautcpd
ncacn-ip-tcp	3062/tcp	ncacn-ip-tcp
ncacn-ip-tcp	3062/udp	ncacn-ip-tcp
ncadg-ip-udp	3063/tcp	ncadg-ip-udp
ncadg-ip-udp	3063/udp	ncadg-ip-udp
slinterbase	3065/tcp	slinterbase
slinterbase	3065/udp	slinterbase
netattachsdmp	3066/tcp	NETATTACHSDMP
netattachsdmp	3066/udp	NETATTACHSDMP
fjhpjp	3067/tcp	FJHPJP
fjhpjp	3067/udp	FJHPJP
ls3bcast	3068/tcp	ls3 Broadcast
ls3bcast	3068/udp	ls3 Broadcast
ls3	3069/tcp	ls3
ls3	3069/udp	ls3
mgxswitch	3070/tcp	MGXSWITCH
mgxswitch	3070/udp	MGXSWITCH
#	3071-3074	Unassigned
orbix-locator	3075/tcp	Orbix 2000
orbix-locator	3075/udp	Orbix 2000
orbix-config	3076/tcp	Orbix 2000
orbix-config	3076/udp	Orbix 2000
orbix-loc-ssl	3077/tcp	Orbix 2000 SSL
orbix-loc-ssl	3077/udp	Orbix 2000 SSL
orbix-cfg-ssl	3078/tcp	Orbix 2000 SSL
orbix-cfg-ssl	3078/udp	Orbix 2000 SSL
lv-frontpanel	3079/tcp	LV Front Panel
lv-frontpanel	3079/udp	LV Front Panel
stm_pproc	3080/tcp	stm_pproc
stm_pproc	3080/udp	stm_pproc
tl1-lv	3081/tcp	TL1-LV
tl1-lv	3081/udp	TL1-LV
tl1-raw	3082/tcp	TL1-RAW
tl1-raw	3082/udp	TL1-RAW
tl1-telnet	3083/tcp	TL1-TELNET
tl1-telnet	3083/udp	TL1-TELNET
itm-mccs	3084/tcp	ITM-MCCS
itm-mccs	3084/udp	ITM-MCCS
pcihreq	3085/tcp	PCIHReq
pcihreq	3085/udp	PCIHReq
jdl-dbkitchen	3086/tcp	JDL-DBKitchen
jdl-dbkitchen	3086/udp	JDL-DBKitchen
#	3084-3104	Unassigned
cardbox	3105/tcp	Cardbox
cardbox	3105/udp	Cardbox
cardbox-http	3106/tcp	Cardbox HTTP
cardbox-http	3106/udp	Cardbox HTTP
#	3107-3129	Unassigned
icpv2	3130/tcp	ICPv2
icpv2	3130/udp	ICPv2
netbookmark	3131/tcp	Net Book Mark
netbookmark	3131/udp	Net Book Mark
#	3132-3140	Unassigned
vmodem	3141/tcp	VMODEM
vmodem	3141/udp	VMODEM
rdc-wh-eos	3142/tcp	RDC WH EOS
rdc-wh-eos	3142/udp	RDC WH EOS
seaview	3143/tcp	Sea View

lontalk-urgnt	1629/udp	LonTalk urgent
oraclenet8cman	1630/tcp	Oracle Net8 Cman
oraclenet8cman	1630/udp	Oracle Net8 Cman
visitview	1631/tcp	Visit view
visitview	1631/udp	Visit view
pammratc	1632/tcp	PAMMRATC
pammratc	1632/udp	PAMMRATC
pammrpc	1633/tcp	PAMMRPC
pammrpc	1633/udp	PAMMRPC
loaprobe	1634/tcp	America Probe
loaprobe	1634/udp	America Probe
edb-server1	1635/tcp	EDB Server 1
edb-server1	1635/udp	EDB Server 1
cncp	1636/tcp	CableNet
cncp	1636/udp	CableNet
cnap	1637/tcp	CableNet Admin
cnap	1637/udp	CableNet Admin
cnip	1638/tcp	CableNet Info
cnip	1638/udp	CableNet Info
cert-initiator	1639/tcp	cert-initiator
cert-initiator	1639/udp	cert-initiator
cert-responder	1640/tcp	cert-responder
cert-responder	1640/udp	cert-responder
invision	1641/tcp	InVision
invision	1641/udp	InVision
isis-am	1642/tcp	isis-am
isis-am	1642/udp	isis-am
isis-ambc	1643/tcp	isis-ambc
isis-ambc	1643/udp	isis-ambc
saiseh	1644/tcp	Satellite-data 4
datametrics	1645/tcp	datametrics
datametrics	1645/udp	datametrics
sa-msg-port	1646/tcp	sa-msg-port
sa-msg-port	1646/udp	sa-msg-port
rsap	1647/tcp	rsap
rsap	1647/udp	rsap
concurrent-lm	1648/tcp	concurrent-lm
concurrent-lm	1648/udp	concurrent-lm
kermit	1649/tcp	kermit
kermit	1649/udp	kermit
nkd	1650/tcp	nkd
nkd	1650/udp	nkd
shiva_confsrvr	1651/tcp	shiva_confsrvr
shiva_confsrvr	1651/udp	shiva_confsrvr
xnmp	1652/tcp	xnmp
xnmp	1652/udp	xnm
alphatech-lm	1653/tcp	alphatech-lm
alphatech-lm	1653/udp	alphatech-lm
stargatealerts	1654/tcp	stargatealerts
stargatealerts	1654/udp	stargatealerts
dec-mbadmin	1655/tcp	dec-mbadmin
dec-mbadmin	1655/udp	dec-mbadmin
dec-mbadmin-h	1656/tcp	dec-mbadmin-h
dec-mbadmin-h	1656/udp	dec-mbadmin-h
fujitsu-mmpdc	1657/tcp	fujitsu-mmpdc
fujitsu-mmpdc	1657/udp	fujitsu-mmpdc
sixnetudr	1658/tcp	sixnetudr
sixnetudr	1658/udp	sixnetudr
sg-lm	1659/tcp	Silicon Grail
sg-lm	1659/udp	Silicon Grail
skip-mc-gikreq	1660/tcp	skip-mc-gikreq
skip-mc-gikreq	1660/udp	skip-mc-gikreq
netview-aix-1	1661/tcp	netview-aix-1
netview-aix-1	1661/udp	netview-aix-1
netview-aix-2	1662/tcp	netview-aix-2
netview-aix-2	1662/udp	netview-aix-2
netview-aix-3	1663/tcp	netview-aix-3
netview-aix-3	1663/udp	netview-aix-3
netview-aix-4	1664/tcp	netview-aix-4
netview-aix-4	1664/udp	netview-aix-4
netview-aix-5	1665/tcp	netview-aix-5
seaview	3143/udp	Sea View
tarantella	3144/tcp	Tarantella
tarantella	3144/udp	Tarantella
csi-lfap	3145/tcp	CSI-LFAP
csi-lfap	3145/udp	CSI-LFAP
#	3146	Unassigned
rfio	3147/tcp	RFIO
rfio	3147/udp	RFIO
nm-game-admin	3148/tcp	NetMike Game
nm-game-admin	3148/udp	NetMike Game
nm-game-server	3149/tcp	NetMike Game
nm-game-server	3149/udp	NetMike Game
nm-asses-admin	3150/tcp	NetMike Assessor
nm-asses-admin	3150/udp	NetMike Assessor
nm-assessor	3151/tcp	NetMike
nm-assessor	3151/udp	NetMike
#	3152-3179	Unassigned
mc-brk-srv	3180/tcp	Millicent Broker
mc-brk-srv	3180/udp	Millicent Broker
bmcpatrolagent	3181/tcp	BMC Patrol Agent
bmcpatrolagent	3181/udp	BMC Patrol Agent
bmcpatrolrnvu	3182/tcp	BMC Patrol
bmcpatrolrnvu	3182/udp	BMC Patrol
#	3183-3261	Unassigned
necp	3262/tcp	NECP
necp	3262/udp	NECP
#	3263	Unassigned
ccmail	3264/tcp	cc:mail/lotus
ccmail	3264/udp	cc:mail/lotus
altav-tunnel	3265/tcp	Altav Tunnel
altav-tunnel	3265/udp	Altav Tunnel
ns-cfg-server	3266/tcp	NS CFG Server
ns-cfg-server	3266/udp	NS CFG Server
ibm-dial-out	3267/tcp	IBM Dial Out
ibm-dial-out	3267/udp	IBM Dial Out
msft-gc	3268/tcp	Microsoft Global
msft-gc	3268/udp	Microsoft Global
msft-gc-ssl	3269/tcp	Microsoft Global
msft-gc-ssl	3269/udp	Microsoft Global
verismart	3270/tcp	Verismart
verismart	3270/udp	Verismart
csoft-prev	3271/tcp	CSoft Prev Port
csoft-prev	3271/udp	CSoft Prev Port
user-manager	3272/tcp	Fujitsu User Mgr
user-manager	3272/udp	Fujitsu User Mgr
sxmp	3273/tcp	SXMP
sxmp	3273/udp	SXMP
ordinox-server	3274/tcp	Ordinox Server
ordinox-server	3274/udp	Ordinox Server
samd	3275/tcp	SAMD
samd	3275/udp	SAMD
maxim-asics	3276/tcp	Maxim ASICs
maxim-asics	3276/udp	Maxim ASICs
awg-proxy	3277/tcp	AWG Proxy
awg-proxy	3277/udp	AWG Proxy
lkcmserver	3278/tcp	LKCM Server
lkcmserver	3278/udp	LKCM Server
admind	3279/tcp	admind
admind	3279/udp	admind
vs-server	3280/tcp	VS Server
vs-server	3280/udp	VS Server
sysopt	3281/tcp	SYSOPT
sysopt	3281/udp	SYSOPT
datusorb	3282/tcp	Datusorb
datusorb	3282/udp	Datusorb
net-assistant	3283/tcp	Net Assistant
net-assistant	3283/udp	Net Assistant
4talk	3284/tcp	4Talk
4talk	3284/udp	4Talk
plato	3285/tcp	Plato
plato	3285/udp	Plato

netview-aix-5	1665/udp	netview-aix-5
netview-aix-6	1666/tcp	netview-aix-6
netview-aix-6	1666/udp	netview-aix-6
netview-aix-7	1667/tcp	netview-aix-7
netview-aix-7	1667/udp	netview-aix-7
netview-aix-8	1668/tcp	netview-aix-8
netview-aix-8	1668/udp	netview-aix-8
netview-aix-9	1669/tcp	netview-aix-9
netview-aix-9	1669/udp	netview-aix-9
netview-aix-10	1670/tcp	netview-aix-10
netview-aix-10	1670/udp	netview-aix-10
netview-aix-11	1671/tcp	netview-aix-11
netview-aix-11	1671/udp	netview-aix-11
netview-aix-12	1672/tcp	netview-aix-12
netview-aix-12	1672/udp	netview-aix-12
proshare-mc-1	1673/tcp	Intel Proshare
proshare-mc-1	1673/udp	Intel Proshare
proshare-mc-2	1674/tcp	Intel Proshare
proshare-mc-2	1674/udp	Intel Proshare
pdp	1675/tcp	Pacific Data
pdp	1675/udp	Pacific Data
netcomm1	1676/tcp	netcomm1
netcomm2	1676/udp	netcomm2
groupwise	1677/tcp	groupwise
groupwise	1677/udp	groupwise
prolink	1678/tcp	prolink
prolink	1678/udp	prolink
darcorp-lm	1679/tcp	darcorp-lm
darcorp-lm	1679/udp	darcorp-lm
microcom-sbp	1680/tcp	microcom-sbp
microcom-sbp	1680/udp	microcom-sbp
sd-elmd	1681/tcp	sd-elmd
sd-elmd	1681/udp	sd-elmd
lanyon-lantern	1682/tcp	lanyon-lantern
lanyon-lantern	1682/udp	lanyon-lantern
ncpm-hip	1683/tcp	ncpm-hip
ncpm-hip	1683/udp	ncpm-hip
snaresecure	1684/tcp	SnareSecure
snaresecure	1684/udp	SnareSecure
n2nremote	1685/tcp	n2nremote
n2nremote	1685/udp	n2nremote
cvmon	1686/tcp	cvmon
cvmon	1686/udp	cvmon
nsjtp-ctrl	1687/tcp	nsjtp-ctrl
nsjtp-ctrl	1687/udp	nsjtp-ctrl
nsjtp-data	1688/tcp	nsjtp-data
nsjtp-data	1688/udp	nsjtp-data
firefox	1689/tcp	firefox
firefox	1689/udp	firefox
ng-umds	1690/tcp	ng-umds
ng-umds	1690/udp	ng-umds
empire-empuma	1691/tcp	empire-empuma
empire-empuma	1691/udp	empire-empuma
sstsys-lm	1692/tcp	sstsys-lm
sstsys-lm	1692/udp	sstsys-lm
rrirtr	1693/tcp	rrirtr
rrirtr	1693/udp	rrirtr
rrimwm	1694/tcp	rrimwm
rrimwm	1694/udp	rrimwm
rrilwm	1695/tcp	rrilwm
rrilwm	1695/udp	rrilwm
rrifmm	1696/tcp	rrifmm
rrifmm	1696/udp	rrifmm
rrisat	1697/tcp	rrisat
rrisat	1697/udp	rrisat
rsvp-encap-1	1698/tcp	ENCAPSULATION-1
rsvp-encap-1	1698/udp	ENCAPSULATION-1
rsvp-encap-2	1699/tcp	ENCAPSULATION-2
rsvp-encap-2	1699/udp	ENCAPSULATION-2
mps-raft	1700/tcp	mps-raft
mps-raft	1700/udp	mps-raft
e-net	3286/tcp	E-Net
e-net	3286/udp	E-Net
directvdata	3287/tcp	DIRECTVDATA
directvdata	3287/udp	DIRECTVDATA
cops	3288/tcp	COPS
cops	3288/udp	COPS
enpc	3289/tcp	ENPC
enpc	3289/udp	ENPC
caps-lm	3290/tcp	CAPS LOGISTICS
caps-lm	3290/udp	CAPS LOGISTICS
sah-lm	3291/tcp	S A Holditch &
sah-lm	3291/udp	S A Holditch &
cart-o-rama	3292/tcp	Cart O Rama
cart-o-rama	3292/udp	Cart O Rama
fg-fps	3293/tcp	fg-fps
fg-fps	3293/udp	fg-fps
fg-gip	3294/tcp	fg-gip
fg-gip	3294/udp	fg-gip
dyniplookup	3295/tcp	Dynamic IP
dyniplookup	3295/udp	Dynamic IP
rib-slm	3296/tcp	Rib License Mgr
rib-slm	3296/udp	Rib License Mgr
cytel-lm	3297/tcp	Cytel Mgr
cytel-lm	3297/udp	Cytel Mgr
transview	3298/tcp	Transview
transview	3298/udp	Transview
pdrncs	3299/tcp	pdrncs
pdrncs	3299/udp	pdrncs
mcs-fastmail	3302/tcp	MCS Fastmail
mcs-fastmail	3302/udp	MCS Fastmail
opsession-clnt	3303/tcp	OP Session Clt
opsession-clnt	3303/udp	OP Session Clt
opsession-srvr	3304/tcp	OP Session Serv
opsession-srvr	3304/udp	OP Session Serv
odette-ftp	3305/tcp	ODETTE-FTP
odette-ftp	3305/udp	ODETTE-FTP
mysql	3306/tcp	MySQL
mysql	3306/udp	MySQL
opsession-prxy	3307/tcp	OP Session Proxy
opsession-prxy	3307/udp	OP Session Proxy
tns-server	3308/tcp	TNS Server
tns-server	3308/udp	TNS Server
tns-adv	3309/tcp	TNS ADV
tns-adv	3309/udp	TND ADV
dyna-access	3310/tcp	Dyna Access
dyna-access	3310/udp	Dyna Access
mcns-tel-ret	3311/tcp	MCNS Tel Ret
mcns-tel-ret	3311/udp	MCNS Tel Ret
appman-server	3312/tcp	Application
appman-server	3312/udp	Application
uorb	3313/tcp	Unify Object
uorb	3313/udp	Unify Object
uohost	3314/tcp	Unify Object
uohost	3314/udp	Unify Object
cdid	3315/tcp	CDID
cdid	3315/udp	CDID
aicc-cmi	3316/tcp	AICC/CMI
aicc-cmi	3316/udp	AICC/CMI
vsaiport	3317/tcp	VSAI PORT
vsaiport	3317/udp	VSAI PORT
ssrip	3318/tcp	Swith to Swith
ssrip	3318/udp	Swith to Swith
sdt-lmd	3319/tcp	SDT License Mgr
sdt-lmd	3319/udp	SDT License Mgr
officelink2000	3320/tcp	Office Link 2000
officelink2000	3320/udp	Office Link 2000
vnsstr	3321/tcp	VNSSTR
vnsstr	3321/udp	VNSSTR
active-net	3322-3325	Active Networks
sftu	3326/tcp	SFTU
sftu	3326/udp	SFTU

l2f	1701/tcp	l2f
l2f	1701/udp	l2f
l2tp	1701/tcp	l2tp
l2tp	1701/udp	l2tp
deskshare	1702/tcp	deskshare
deskshare	1702/udp	deskshare
bcs-broker	1704/tcp	bcs-broker
bcs-broker	1704/udp	bcs-broker
slingshot	1705/tcp	slingshot
slingshot	1705/udp	slingshot
jetform	1706/tcp	jetform
jetform	1706/udp	jetform
vdmplay	1707/tcp	vdmplay
vdmplay	1707/udp	vdmplay
gat-lmd	1708/tcp	gat-lmd
gat-lmd	1708/udp	gat-lmd
centra	1709/tcp	centra
centra	1709/udp	centra
impera	1710/tcp	impera
impera	1710/udp	impera
pptconference	1711/tcp	pptconference
pptconference	1711/udp	pptconference
registrar	1712/tcp	resource mon
registrar	1712/udp	resource mon
conferencetalk	1713/tcp	ConferenceTalk
conferencetalk	1713/udp	ConferenceTalk
sesi-lm	1714/tcp	sesi-lm
sesi-lm	1714/udp	sesi-lm
houdini-lm	1715/tcp	houdini-lm
houdini-lm	1715/udp	houdini-lm
xmsg	1716/tcp	xmsg
xmsg	1716/udp	xmsg
fj-hdnet	1717/tcp	fj-hdnet
fj-hdnet	1717/udp	fj-hdnet
h323gatedisc	1718/tcp	h323gatedisc
h323gatedisc	1718/udp	h323gatedisc
h323gatestat	1719/tcp	h323gatestat
h323gatestat	1719/udp	h323gatestat
h323hostcall	1720/tcp	h323hostcall
h323hostcall	1720/udp	h323hostcall
caicci	1721/tcp	caicci
caicci	1721/udp	caicci
hks-lm	1722/tcp	HKS
hks-lm	1722/udp	HKS
pptp	1723/tcp	pptp
pptp	1723/udp	pptp
csbphonemaster	1724/tcp	csbphonemaster
csbphonemaster	1724/udp	csbphonemaster
iden-ralp	1725/tcp	iden-ralp
iden-ralp	1725/udp	iden-ralp
iberiagames	1726/tcp	IBERIAGAMES
iberiagames	1726/udp	IBERIAGAMES
winddx	1727/tcp	winddx
winddx	1727/udp	winddx
telindus	1728/tcp	TELINDUS
telindus	1728/udp	TELINDUS
roketz	1730/tcp	roketz
roketz	1730/udp	roketz
msiccp	1731/tcp	MSICCP
msiccp	1731/udp	MSICCP
proxim	1732/tcp	proxim
proxim	1732/udp	proxim
siipat	1733/tcp	SIMS
siipat	1733/udp	SIMS
cambertx-lm	1734/tcp	Camber
cambertx-lm	1734/udp	Camber
privatechat	1735/tcp	PrivateChat
privatechat	1735/udp	PrivateChat
street-stream	1736/tcp	street-stream
street-stream	1736/udp	street-stream
ultimad	1737/tcp	ultimad
bbars	3327/tcp	BBARS
bbars	3327/udp	BBARS
egptlm	3328/tcp	Eaglepoint
egptlm	3328/udp	Eaglepoint
hp-device-disc	3329/tcp	HP Device Disc
hp-device-disc	3329/udp	HP Device Disc
mcs-calypsoicf	3330/tcp	MCS Calypso ICF
mcs-calypsoicf	3330/udp	MCS Calypso ICF
mcs-messaging	3331/tcp	MCS Messaging
mcs-messaging	3331/udp	MCS Messaging
mcs-mailsvr	3332/tcp	MCS Mail Server
mcs-mailsvr	3332/udp	MCS Mail Server
dec-notes	3333/tcp	DEC Notes
dec-notes	3333/udp	DEC Notes
directv-web	3334/tcp	Direct TV
directv-web	3334/udp	Direct TV
directv-soft	3335/tcp	Direct TV
directv-soft	3335/udp	Direct TV
directv-tick	3336/tcp	Direct TV
directv-tick	3336/udp	Direct TV
directv-catlg	3337/tcp	Direct TV Data
directv-catlg	3337/udp	Direct TV Data
anet-b	3338/tcp	OMF data b
anet-b	3338/udp	OMF data b
anet-l	3339/tcp	OMF data l
anet-l	3339/udp	OMF data l
anet-m	3340/tcp	OMF data m
anet-m	3340/udp	OMF data m
anet-h	3341/tcp	OMF data h
anet-h	3341/udp	OMF data h
webtie	3342/tcp	WebTIE
webtie	3342/udp	WebTIE
ms-cluster-net	3343/tcp	MS Cluster Net
ms-cluster-net	3343/udp	MS Cluster Net
bnt-manager	3344/tcp	BNT Manager
bnt-manager	3344/udp	BNT Manager
influence	3345/tcp	Influence
influence	3345/udp	Influence
trnsprntproxy	3346/tcp	Trnsprnt Proxy
trnsprntproxy	3346/udp	Trnsprnt Proxy
phoenix-rpc	3347/tcp	Phoenix RPC
phoenix-rpc	3347/udp	Phoenix RPC
pangolin-laser	3348/tcp	Pangolin Laser
pangolin-laser	3348/udp	Pangolin Laser
chevinservices	3349/tcp	Chevin Services
chevinservices	3349/udp	Chevin Services
findviatv	3350/tcp	FINDVIATV
findviatv	3350/udp	FINDVIATV
btrieve	3351/tcp	BTRIEVE
btrieve	3351/udp	BTRIEVE
ssql	3352/tcp	SSQL
ssql	3352/udp	SSQL
fatpipe	3353/tcp	FATPIPE
fatpipe	3353/udp	FATPIPE
suitjd	3354/tcp	SUITJD
suitjd	3354/udp	SUITJD
ordinox-dbase	3355/tcp	Ordinox Dbase
ordinox-dbase	3355/udp	Ordinox Dbase
upnotifyps	3356/tcp	UPNOTIFYPS
upnotifyps	3356/udp	UPNOTIFYPS
adtech-test	3357/tcp	Adtech Test IP
adtech-test	3357/udp	Adtech Test IP
mpsysrmsvr	3358/tcp	Mp Sys Rmsvr
mpsysrmsvr	3358/udp	Mp Sys Rmsvr
wg-netforce	3359/tcp	WG NetForce
wg-netforce	3359/udp	WG NetForce
kv-server	3360/tcp	KV Server
kv-server	3360/udp	KV Server
kv-agent	3361/tcp	KV Agent
kv-agent	3361/udp	KV Agent
dj-ilm	3362/tcp	DJ ILM

ultimad	1737/udp	ultimad
gamegen1	1738/tcp	GameGen1
gamegen1	1738/udp	GameGen1
webaccess	1739/tcp	webaccess
webaccess	1739/udp	webaccess
encore	1740/tcp	encore
encore	1740/udp	encore
cisco-net-mgmt	1741/tcp	cisco-net-mgmt
cisco-net-mgmt	1741/udp	cisco-net-mgmt
3Com-nsd	1742/tcp	3Com-nsd
3Com-nsd	1742/udp	3Com-nsd
cinegrfx-lm	1743/tcp	Cinema Graphics
cinegrfx-lm	1743/udp	Cinema Graphics
ncpm-ft	1744/tcp	ncpm-ft
ncpm-ft	1744/udp	ncpm-ft
remote-winsock	1745/tcp	remote-winsock
remote-winsock	1745/udp	remote-winsock
ftrapid-1	1746/tcp	ftrapid-1
ftrapid-1	1746/udp	ftrapid-1
ftrapid-2	1747/tcp	ftrapid-2
ftrapid-2	1747/udp	ftrapid-2
oracle-em1	1748/tcp	oracle-em1
oracle-em1	1748/udp	oracle-em1
aspen-services	1749/tcp	aspen-services
aspen-services	1749/udp	aspen-services
sslp	1750/tcp	Simple Socket
sslp	1750/udp	Simple Socket
swiftnet	1751/tcp	SwiftNet
swiftnet	1751/udp	SwiftNet
lofr-lm	1752/tcp	Leap of Faith
lofr-lm	1752/udp	Leap of Faith
translogic-lm	1753/tcp	Translogic
translogic-lm	1753/udp	Translogic
oracle-em2	1754/tcp	oracle-em2
oracle-em2	1754/udp	oracle-em2
ms-streaming	1755/tcp	ms-streaming
ms-streaming	1755/udp	ms-streaming
capfast-lmd	1756/tcp	capfast-lmd
capfast-lmd	1756/udp	capfast-lmd
cnhrp	1757/tcp	cnhrp
cnhrp	1757/udp	cnhrp
tftp-mcast	1758/tcp	tftp-mcast
tftp-mcast	1758/udp	tftp-mcast
spss-lm	1759/tcp	SPSS
spss-lm	1759/udp	SPSS
www-ldap-gw	1760/tcp	www-ldap-gw
www-ldap-gw	1760/udp	www-ldap-gw
cft-0	1761/tcp	cft-0
cft-0	1761/udp	cft-0
cft-1	1762/tcp	cft-1
cft-1	1762/udp	cft-1
cft-2	1763/tcp	cft-2
cft-2	1763/udp	cft-2
cft-3	1764/tcp	cft-3
cft-3	1764/udp	cft-3
cft-4	1765/tcp	cft-4
cft-4	1765/udp	cft-4
cft-5	1766/tcp	cft-5
cft-5	1766/udp	cft-5
cft-6	1767/tcp	cft-6
cft-6	1767/udp	cft-6
cft-7	1768/tcp	cft-7
cft-7	1768/udp	cft-7
bmc-net-adm	1769/tcp	bmc-net-adm
bmc-net-adm	1769/udp	bmc-net-adm
bmc-net-svc	1770/tcp	bmc-net-svc
bmc-net-svc	1770/udp	bmc-net-svc
vaultbase	1771/tcp	vaultbase
vaultbase	1771/udp	vaultbase
essweb-gw	1772/tcp	EssWeb Gateway
essweb-gw	1772/udp	EssWeb Gateway
dj-ilm	3362/udp	DJ ILM
nati-vi-server	3363/tcp	NATI Vi Server
nati-vi-server	3363/udp	NATI Vi Server
creativeserver	3364/tcp	Creative Server
creativeserver	3364/udp	Creative Server
contentserver	3365/tcp	Content Server
contentserver	3365/udp	Content Server
creativepartnr	3366/tcp	Creative Partner
creativepartnr	3366/udp	Creative Partner
satvid-datalnk	3367-3371	Satellite Video
tip2	3372/tcp	TIP 2
tip2	3372/udp	TIP 2
lavenir-lm	3373/tcp	Lavenir
lavenir-lm	3373/udp	Lavenir
cluster-disc	3374/tcp	Cluster Disc
cluster-disc	3374/udp	Cluster Disc
vsnm-agent	3375/tcp	VSNM Agent
vsnm-agent	3375/udp	VSNM Agent
cdborker	3376/tcp	CD Broker
cdbroker	3376/udp	CD Broker
cogsys-lm	3377/tcp	Cogsys Network
cogsys-lm	3377/udp	Cogsys Network
wsicopy	3378/tcp	WSICOPY
wsicopy	3378/udp	WSICOPY
socorfs	3379/tcp	SOCORFS
socorfs	3379/udp	SOCORFS
sns-channels	3380/tcp	SNS Channels
sns-channels	3380/udp	SNS Channels
geneous	3381/tcp	Geneous
geneous	3381/udp	Geneous
fujitsu-neat	3382/tcp	Fujitsu Network
fujitsu-neat	3382/udp	Fujitsu Network
esp-lm	3383/tcp	Enterprise
esp-lm	3383/udp	Enterprise
hp-clic	3384/tcp	Cluster
hp-clic	3384/udp	Hardware
qnxnetman	3385/tcp	qnxnetman
qnxnetman	3385/udp	qnxnetman
gprs-data	3386/tcp	GPRS Data
gprs-sig	3386/udp	GPRS SIG
backroomnet	3387/tcp	Back Room Net
backroomnet	3387/udp	Back Room Net
cbserver	3388/tcp	CB Server
cbserver	3388/udp	CB Server
ms-wbt-server	3389/tcp	MS WBT Server
ms-wbt-server	3389/udp	MS WBT Server
dsc	3390/tcp	Distributed
dsc	3390/udp	Distributed
savant	3391/tcp	SAVANT
savant	3391/udp	SAVANT
efi-lm	3392/tcp	EFI License
efi-lm	3392/udp	EFI License
d2k-tapestry1	3393/tcp	D2K Tapestry
d2k-tapestry1	3393/udp	D2K Tapestry
d2k-tapestry2	3394/tcp	D2K Tapestry
d2k-tapestry2	3394/udp	D2K Tapestry
dyna-lm	3395/tcp	Dyna (Elam)
dyna-lm	3395/udp	Dyna (Elam)
printer_agent	3396/tcp	Printer Agent
printer_agent	3396/udp	Printer Agent
cloanto-lm	3397/tcp	Cloanto
cloanto-lm	3397/udp	Cloanto
mercantile	3398/tcp	Mercantile
mercantile	3398/udp	Mercantile
csms	3399/tcp	CSMS
csms	3399/udp	CSMS
csms2	3400/tcp	CSMS2
csms2	3400/udp	CSMS2
filecast	3401/tcp	filecast
filecast	3401/udp	filecast
#	3402-3420	Unassigned

kmscontrol	1773/tcp	KMSControl
kmscontrol	1773/udp	KMSControl
global-dtserv	1774/tcp	global-dtserv
global-dtserv	1774/udp	global-dtserv
#	1775/tcp	
femis	1776/tcp	F E M I S
femis	1776/udp	F E M I S
powerguardian	1777/tcp	powerguardian
powerguardian	1777/udp	powerguardian
prodigy-intrnet	1778/tcp	prodigy
prodigy-intrnet	1778/udp	prodigy
pharmasoft	1779/tcp	pharmasoft
pharmasoft	1779/udp	pharmasoft
dpkeyserv	1780/tcp	dpkeyserv
dpkeyserv	1780/udp	dpkeyserv
answersoft-lm	1781/tcp	answersoft-lm
answersoft-lm	1781/udp	answersoft-lm
hp-hcip	1782/tcp	hp-hcip
hp-hcip	1782/udp	hp-hcip
#	1783	Decomissioned P
finle-lm	1784/tcp	Finle
finle-lm	1784/udp	Finle
windlm	1785/tcp	Wind River
windlm	1785/udp	Wind River
funk-logger	1786/tcp	funk-logger
funk-logger	1786/udp	funk-logger
funk-license	1787/tcp	funk-license
funk-license	1787/udp	funk-license
psmond	1788/tcp	psmond
psmond	1788/udp	psmond
hello	1789/tcp	hello
hello	1789/udp	hello
nmsp	1790/tcp	Narrative Media
nmsp	1790/udp	Narrative Media
ea1	1791/tcp	EA1
ea1	1791/udp	EA1
ibm-dt-2	1792/tcp	ibm-dt-2
ibm-dt-2	1792/udp	ibm-dt-2
rsc-robot	1793/tcp	rsc-robot
rsc-robot	1793/udp	rsc-robot
cera-bcm	1794/tcp	cera-bcm
cera-bcm	1794/udp	cera-bcm
dpi-proxy	1795/tcp	dpi-proxy
dpi-proxy	1795/udp	dpi-proxy
vocaltec-admin	1796/tcp	Vocaltec Server
vocaltec-admin	1796/udp	Vocaltec Server
uma	1797/tcp	UMA
uma	1797/udp	UMA
etp	1798/tcp	Event Transfer
etp	1798/udp	Event Transfer
netrisk	1799/tcp	NETRISK
netrisk	1799/udp	NETRISK
ansys-lm	1800/tcp	ANSYS
ansys-lm	1800/udp	ANSYS
msmq	1801/tcp	MS Message Que
msmq	1801/udp	MS Message Que
concomp1	1802/tcp	ConComp1
concomp1	1802/udp	ConComp1
hp-hcip-gwy	1803/tcp	HP-HCIP-GWY
hp-hcip-gwy	1803/udp	HP-HCIP-GWY
enl	1804/tcp	ENL
enl	1804/udp	ENL
enl-name	1805/tcp	ENL-Name
enl-name	1805/udp	ENL-Name
musiconline	1806/tcp	Musiconline
musiconline	1806/udp	Musiconline
fhsp	1807/tcp	Fujitsu Hot
fhsp	1807/udp	Fujitsu Hot
oracle-vp2	1808/tcp	Oracle-VP2
oracle-vp2	1808/udp	Oracle-VP2
oracle-vp1	1809/tcp	Oracle-VP1
bmap	3421/tcp	Bull Apprise
bmap	3421/udp	Bull Apprise
#	3422-3453	Unassigned
mira	3454/tcp	Apple Remote
prsvp	3455/tcp	RSVP Port
prsvp	3455/udp	RSVP Port
vat	3456/tcp	VAT default data
vat	3456/udp	VAT default data
vat-control	3457/tcp	VAT default Ctrl
vat-control	3457/udp	VAT default Ctrl
d3winosfi	3458/tcp	D3WinOsfi
d3winosfi	3458/udp	DsWinOSFI
integral	3459/tcp	TIP Integral
integral	3459/udp	TIP Integral
edm-manager	3460/tcp	EDM Manger
edm-manager	3460/udp	EDM Manger
edm-stager	3461/tcp	EDM Stager
edm-stager	3461/udp	EDM Stager
edm-std-notify	3462/tcp	EDM STD Notify
edm-std-notify	3462/udp	EDM STD Notify
edm-adm-notify	3463/tcp	EDM ADM Notify
edm-adm-notify	3463/udp	EDM ADM Notify
edm-mgr-sync	3464/tcp	EDM MGR Sync
edm-mgr-sync	3464/udp	EDM MGR Sync
edm-mgr-cntrl	3465/tcp	EDM MGR Cntrl
edm-mgr-cntrl	3465/udp	EDM MGR Cntrl
workflow	3466/tcp	WORKFLOW
workflow	3466/udp	WORKFLOW
rcst	3467/tcp	RCST
rcst	3467/udp	RCST
ttcmremotectrl	3468/tcp	TTCM Remote Ctrl
ttcmremotectrl	3468/udp	TTCM Remote Ctrl
pluribus	3469/tcp	Pluribus
pluribus	3469/udp	Pluribus
jt400	3470/tcp	jt400
jt400	3470/udp	jt400
jt400-ssl	3471/tcp	jt400-ssl
jt400-ssl	3471/udp	jt400-ssl
#	3472-3534	Unassigned
ms-la	3535/tcp	MS-LA
ms-la	3535/udp	MS-LA
#	3536-3562	Unassigned
watcomdebug	3563/tcp	Watcom Debug
watcomdebug	3563/udp	Watcom Debug
#	3564-3671	Unassigned
harlequinorb	3672/tcp	harlequinorb
harlequinorb	3672/udp	harlequinorb
#	3673-3801	Unassigned
vhd	3802/tcp	VHD
vhd	3802/udp	VHD
#	3803-3844	Unassigned
v-one-spp	3845/tcp	V-ONE Single
v-one-spp	3845/udp	V-ONE Single
#	3846-3861	Unassigned
giga-pocket	3862/tcp	GIGA-POCKET
giga-pocket	3862/udp	GIGA-POCKET
#	3863-3874	Unassigned
pnbscada	3875/tcp	PNBSCADA
pnbscada	3875/udp	PNBSCADA
#	3876-3899	Unassigned
udt_os	3900/tcp	Unidata UDT OS
udt_os	3900/udp	Unidata UDT OS
#	3901-3983	Unassigned
mapper-nodemgr	3984/tcp	MAPPER network
mapper-nodemgr	3984/udp	MAPPER network
mapper-mapethd	3985/tcp	MAPPER TCP/IP
mapper-mapethd	3985/udp	MAPPER TCP/IP
mapper-ws_ethd	3986/tcp	MAPPER
mapper-ws_ethd	3986/udp	MAPPER
centerline	3987/tcp	Centerline
centerline	3987/udp	Centerline

oracle-vp1	1809/udp	Oracle-VP1
jerand-lm	1810/tcp	Jerand
jerand-lm	1810/udp	Jerand
scientia-sdb	1811/tcp	Scientia-SDB
scientia-sdb	1811/udp	Scientia-SDB
radius	1812/tcp	RADIUS
radius	1812/udp	RADIUS
radius-acct	1813/tcp	RADIUS Acc
radius-acct	1813/udp	RADIUS Acc
tdp-suite	1814/tcp	TDP Suite
tdp-suite	1814/udp	TDP Suite
mmpft	1815/tcp	MMPFT
mmpft	1815/udp	MMPFT
harp	1816/tcp	HARP
harp	1816/udp	HARP
rkb-oscs	1817/tcp	RKB-OSCS
rkb-oscs	1817/udp	RKB-OSCS
etftp	1818/tcp	Enhanced TFTP
etftp	1818/udp	Enhanced TFTP
plato-lm	1819/tcp	Plato
plato-lm	1819/udp	Plato
mcagent	1820/tcp	mcagent
mcagent	1820/udp	mcagent
donnyworld	1821/tcp	donnyworld
donnyworld	1821/udp	donnyworld
es-elmd	1822/tcp	es-elmd
es-elmd	1822/udp	es-elmd
unisys-lm	1823/tcp	Unisys
unisys-lm	1823/udp	Unisys
metrics-pas	1824/tcp	metrics-pas
metrics-pas	1824/udp	metrics-pas
direcpc-video	1825/tcp	DirecPC Video
direcpc-video	1825/udp	DirecPC Video
ardt	1826/tcp	ARDT
ardt	1826/udp	ARDT
asi	1827/tcp	ASI
asi	1827/udp	ASI
itm-mcell-u	1828/tcp	itm-mcell-u
itm-mcell-u	1828/udp	itm-mcell-u
optika-emedia	1829/tcp	Optika eMedia
optika-emedia	1829/udp	Optika eMedia
net8-cman	1830/tcp	Oracle Net8
net8-cman	1830/udp	Oracle Net8
myrtle	1831/tcp	Myrtle
myrtle	1831/udp	Myrtle
tht-treasure	1832/tcp	ThoughtTreasure
tht-treasure	1832/udp	ThoughtTreasure
udpradio	1833/tcp	udpradio
udpradio	1833/udp	udpradio
ardusuni	1834/tcp	ARDUS Unicast
ardusuni	1834/udp	ARDUS Unicast
ardusmul	1835/tcp	ARDUS Multicast
ardusmul	1835/udp	ARDUS Multicast
ste-smsc	1836/tcp	ste-smsc
ste-smsc	1836/udp	ste-smsc
csoft1	1837/tcp	csoft1
csoft1	1837/udp	csoft1
talnet	1838/tcp	TALNET
talnet	1838/udp	TALNET
netopia-vo1	1839/tcp	netopia-vo1
netopia-vo1	1839/udp	netopia-vo1
netopia-vo2	1840/tcp	netopia-vo2
netopia-vo2	1840/udp	netopia-vo2
netopia-vo3	1841/tcp	netopia-vo3
netopia-vo3	1841/udp	netopia-vo3
netopia-vo4	1842/tcp	netopia-vo4
netopia-vo4	1842/udp	netopia-vo4
netopia-vo5	1843/tcp	netopia-vo5
netopia-vo5	1843/udp	netopia-vo5
direcpc-dll	1844/tcp	DirecPC-DLL
direcpc-dll	1844/udp	DirecPC-DLL

#	3988-3999	Unassigned
terabase	4000/tcp	Terabase
terabase	4000/udp	Terabase
newoak	4001/tcp	NewOak
newoak	4001/udp	NewOak
pxc-spvr-ft	4002/tcp	pxc-spvr-ft
pxc-spvr-ft	4002/udp	pxc-spvr-ft
pxc-splr-ft	4003/tcp	pxc-splr-ft
pxc-splr-ft	4003/udp	pxc-splr-ft
pxc-roid	4004/tcp	pxc-roid
pxc-roid	4004/udp	pxc-roid
pxc-pin	4005/tcp	pxc-pin
pxc-pin	4005/udp	pxc-pin
pxc-spvr	4006/tcp	pxc-spvr
pxc-spvr	4006/udp	pxc-spvr
pxc-splr	4007/tcp	pxc-splr
pxc-splr	4007/udp	pxc-splr
netcheque	4008/tcp	NetCheque acc
netcheque	4008/udp	NetCheque acc
chimera-hwm	4009/tcp	Chimera HWM
chimera-hwm	4009/udp	Chimera HWM
samsung-unidex	4010/tcp	Samsung Unidex
samsung-unidex	4010/udp	Samsung Unidex
altserviceboot	4011/tcp	Alternate Boot
altserviceboot	4011/udp	Alternate Boot
pda-gate	4012/tcp	PDA Gate
pda-gate	4012/udp	PDA Gate
acl-manager	4013/tcp	ACL Manager
acl-manager	4013/udp	ACL Manager
taiclock	4014/tcp	TAICLOCK
taiclock	4014/udp	TAICLOCK
talarian-mcast1	4015/tcp	Talarian Mcast
talarian-mcast1	4015/udp	Talarian Mcast
talarian-mcast2	4016/tcp	Talarian Mcast
talarian-mcast2	4016/udp	Talarian Mcast
talarian-mcast3	4017/tcp	Talarian Mcast
talarian-mcast3	4017/udp	Talarian Mcast
talarian-mcast4	4018/tcp	Talarian Mcast
talarian-mcast4	4018/udp	Talarian Mcast
talarian-mcast5	4019/tcp	Talarian Mcast
talarian-mcast5	4019/udp	Talarian Mcast
#	4020-4095	Unassigned
bre	4096/tcp	BRE
bre	4096/udp	BRE
patrolview	4097/tcp	Patrol View
patrolview	4097/udp	Patrol View
drmsfsd	4098/tcp	drmsfsd
drmsfsd	4098/udp	drmsfsd
dpcp	4099/tcp	DPCP
dpcp	4099/udp	DPCP
#	4100-4131	Unassigned
nuts_dem	4132/tcp	NUTS Daemon
nuts_dem	4132/udp	NUTS Daemon
nuts_bootp	4133/tcp	NUTS Bootp Serv
nuts_bootp	4133/udp	NUTS Bootp Serv
nifty-hmi	4134/tcp	NIFTY-Serve HMI
nifty-hmi	4134/udp	NIFTY-Serve HMI
oirtgsvc	4141/tcp	Workflow Server
oirtgsvc	4141/udp	Workflow Server
oidocsvc	4142/tcp	Document Server
oidocsvc	4142/udp	Document Server
oidsr	4143/tcp	Document Replic
oidsr	4143/udp	Document Replic
#	4144-4159	Unassigned
jini-discovery	4160/tcp	Jini Discovery
jini-discovery	4160/udp	Jini Discovery
#	4161-4198	Unassigned
eims-admin	4199/tcp	EIMS ADMIN
eims-admin	4199/udp	EIMS ADMIN
vrml-multi-use	4200-4299	VRML Multi
corelccam	4300/tcp	Corel CCam

#	1845-1849	Unassigned
gsi	1850/tcp	GSI
gsi	1850/udp	GSI
ctcd	1851/tcp	ctcd
ctcd	1851/udp	ctcd
#	1852-1859	Unassigned
sunscalar-svc	1860/tcp	SunSCALAR
sunscalar-svc	1860/udp	SunSCALAR
lecroy-vicp	1861/tcp	LeCroy VICP
lecroy-vicp	1861/udp	LeCroy VICP
techra-server	1862/tcp	techra-server
techra-server	1862/udp	techra-server
msnp	1863/tcp	MSNP
msnp	1863/udp	MSNP
paradym-31port	1864/tcp	Paradym 31 Port
paradym-31port	1864/udp	Paradym 31 Port
entp	1865/tcp	ENTP
entp	1865/udp	ENTP
#	1866-1869	Unassigned
sunscalar-dns	1870/tcp	SunSCALAR DNS
sunscalar-dns	1870/udp	SunSCALAR DNS
canocentral0	1871/tcp	Cano Central 0
canocentral0	1871/udp	Cano Central 0
canocentral1	1872/tcp	Cano Central 1
canocentral1	1872/udp	Cano Central 1
fjmpjps	1873/tcp	Fjmpjps
fjmpjps	1873/udp	Fjmpjps
fjswapsnp	1874/tcp	Fjswapsnp
fjswapsnp	1874/udp	Fjswapsnp
#	1875-1880	Unassigned
ibm-mqseries2	1881/tcp	IBM MQSeries
ibm-mqseries2	1881/udp	IBM MQSeries
#	1882-1894	Unassigned
vista-4gl	1895/tcp	Vista 4GL
vista-4gl	1895/udp	Vista 4GL
#	1896-1898	Unassigned
mc2studios	1899/tcp	MC2Studios
mc2studios	1899/udp	MC2Studio
ssdp	1900/tcp	SSDP
ssdp	1900/udp	SSDP
fjicl-tep-a	1901/tcp	Fujitsu ICL A
fjicl-tep-a	1901/udp	Fujitsu ICL A
fjicl-tep-b	1902/tcp	Fujitsu ICL B
fjicl-tep-b	1902/udp	Fujitsu ICL B
linkname	1903/tcp	Local Link Name
linkname	1903/udp	Local Link Name
fjicl-tep-c	1904/tcp	Fujitsu ICL C
fjicl-tep-c	1904/udp	Fujitsu ICL C
sugp	1905/tcp	Secure UP.Link
sugp	1905/udp	Secure UP.Link
tpmd	1906/tcp	TPortMapperReq
tpmd	1906/udp	TPortMapperReq
intrastar	1907/tcp	IntraSTAR
intrastar	1907/udp	IntraSTAR
dawn	1908/tcp	Dawn
dawn	1908/udp	Dawn
global-wlink	1909/tcp	Global World
global-wlink	1909/udp	Global World
ultrabac	1910/tcp	ultrabac
ultrabac	1910/udp	ultrabac
mtp	1911/tcp	Starlight
mtp	1911/udp	Starlight
rhp-iibp	1912/tcp	rhp-iibp
rhp-iibp	1912/udp	rhp-iibp
armadp	1913/tcp	armadp
armadp	1913/udp	armadp
elm-momentum	1914/tcp	Elm-Momentum
elm-momentum	1914/udp	Elm-Momentum
facelink	1915/tcp	FACELINK
facelink	1915/udp	FACELINK
persona	1916/tcp	Persoft Persona
corelccam	4300/udp	Corel CCam
#	4301-4320	Unassigned
rwhois	4321/tcp	Remote Who Is
rwhois	4321/udp	Remote Who Is
unicall	4343/tcp	UNICALL
unicall	4343/udp	UNICALL
vinainstall	4344/tcp	VinaInstall
vinainstall	4344/udp	VinaInstall
m4-network-as	4345/tcp	Macro 4 Network
m4-network-as	4345/udp	Macro 4 Network
elanlm	4346/tcp	ELAN LM
elanlm	4346/udp	ELAN LM
lansurveyor	4347/tcp	LAN Surveyor
lansurveyor	4347/udp	LAN Surveyor
itose	4348/tcp	ITOSE
itose	4348/udp	ITOSE
fsportmap	4349/tcp	FileSys Port Map
fsportmap	4349/udp	FileSys Port Map
net-device	4350/tcp	Net Device
net-device	4350/udp	Net Device
plcy-net-svcs	4351/tcp	PLCY Net Serv
plcy-net-svcs	4351/udp	PLCY Net Serv
#	4352	Unassigned
f5-iquery	4353/tcp	F5 iQuery
f5-iquery	4353/udp	F5 iQuery
#	4354-4443	Unassigned
saris	4442/tcp	Saris
saris	4442/udp	Saris
pharos	4443/tcp	Pharos
pharos	4443/udp	Pharos
krb524	4444/tcp	KRB524
krb524	4444/udp	KRB524
nv-video	4444/tcp	NV Video default
nv-video	4444/udp	NV Video default
upnotifyp	4445/tcp	UPNOTIFYP
upnotifyp	4445/udp	UPNOTIFYP
n1-fwp	4446/tcp	N1-FWP
n1-fwp	4446/udp	N1-FWP
n1-rmgmt	4447/tcp	N1-RMGMT
n1-rmgmt	4447/udp	N1-RMGMT
asc-slmd	4448/tcp	ASC Licence Mgr
asc-slmd	4448/udp	ASC Licence Mgr
privatewire	4449/tcp	PrivateWire
privatewire	4449/udp	PrivateWire
camp	4450/tcp	Camp
camp	4450/udp	Camp
ctisystemmsg	4451/tcp	CTI System Msg
ctisystemmsg	4451/udp	CTI System Msg
ctiprogramload	4452/tcp	CTI Program Load
ctiprogramload	4452/udp	CTI Program Load
nssalertmgr	4453/tcp	NSS Alert Mgr
nssalertmgr	4453/udp	NSS Alert Mgr
nssagentmgr	4454/tcp	NSS Agent Mgr
nssagentmgr	4454/udp	NSS Agent Mgr
prchat-user	4455/tcp	PR Chat User
prchat-user	4455/udp	PR Chat User
prchat-server	4456/tcp	PR Chat Server
prchat-server	4456/udp	PR Chat Server
prRegister	4457/tcp	PR Register
prRegister	4457/udp	PR Register
#	4458-4499	Unassigned
sae-urn	4500/tcp	sae-urn
sae-urn	4500/udp	sae-urn
urn-x-cdchoice	4501/tcp	urn-x-cdchoice
urn-x-cdchoice	4501/udp	urn-x-cdchoice
worldscores	4545/tcp	WorldScores
worldscores	4545/udp	WorldScores
sf-lm	4546/tcp	SF (Sentinel)
sf-lm	4546/udp	SF (Sentinel)
lanner-lm	4547/tcp	Lanner
lanner-lm	4547/udp	Lanner

persona	1916/udp	Persoft Persona
noagent	1917/tcp	nOAgent
noagent	1917/udp	nOAgent
can-nds	1918/tcp	Candle NDS
can-nds	1918/udp	Candle NDS
can-dch	1919/tcp	Candle DCH
can-dch	1919/udp	Candle DCH
can-ferret	1920/tcp	Candle FERRET
can-ferret	1920/udp	Candle FERRET
noadmin	1921/tcp	NoAdmin
noadmin	1921/udp	NoAdmin
tapestry	1922/tcp	Tapestry
tapestry	1922/udp	Tapestry
spice	1923/tcp	SPICE
spice	1923/udp	SPICE
xiip	1924/tcp	XIIP
xiip	1924/udp	XIIP
#	1925-1929	Unassigned
driveappserver	1930/tcp	Drive AppServer
driveappserver	1930/udp	Drive AppServer
amdsched	1931/tcp	AMD SCHED
amdsched	1931/udp	AMD SCHED
#	1932-1943	Unassigned
close-combat	1944/tcp	close-combat
close-combat	1944/udp	close-combat
dialogic-elmd	1945/tcp	dialogic-elmd
dialogic-elmd	1945/udp	dialogic-elmd
tekpls	1946/tcp	tekpls
tekpls	1946/udp	tekpls
hlserver	1947/tcp	hlserver
hlserver	1947/udp	hlserver
eye2eye	1948/tcp	eye2eye
eye2eye	1948/udp	eye2eye
ismaeasdaqlive	1949/tcp	ISMA Easdaq Live
ismaeasdaqlive	1949/udp	ISMA Easdaq Live
ismaeasdaqtest	1950/tcp	ISMA Easdaq Test
ismaeasdaqtest	1950/udp	ISMA Easdaq Test
bcs-lmserver	1951/tcp	bcs-lmserver
bcs-lmserver	1951/udp	bcs-lmserver
mpnjsc	1952/tcp	mpnjsc
mpnjsc	1952/udp	mpnjsc
rapidbase	1953/tcp	Rapid Base
rapidbase	1953/udp	Rapid Base
#	1954-1960	Unassigned
bts-appserver	1961/tcp	BTS APPSERVER
bts-appserver	1961/udp	BTS APPSERVER
biap-mp	1962/tcp	BIAP-MP
biap-mp	1962/udp	BIAP-MP
webmachine	1963/tcp	WebMachine
webmachine	1963/udp	WebMachine
solid-e-engine	1964/tcp	SOLID E ENGINE
solid-e-engine	1964/udp	SOLID E ENGINE
tivoli-npm	1965/tcp	Tivoli NPM
tivoli-npm	1965/udp	Tivoli NPM
slush	1966/tcp	Slush
slush	1966/udp	Slush
sns-quote	1967/tcp	SNS Quote
sns-quote	1967/udp	SNS Quote
#	1968-1971	Unassigned
intersys-cache	1972/tcp	Cache
intersys-cache	1972/udp	Cache
dlsrap	1973/tcp	Data Link
dlsrap	1973/udp	Data Link
drp	1974/tcp	DRP
drp	1974/udp	DRP
tcoflashagent	1975/tcp	TCO Flash Agent
tcoflashagent	1975/udp	TCO Flash Agent
tcoregagent	1976/tcp	TCO Reg Agent
tcoregagent	1976/udp	TCO Reg Agent
tcoaddressbook	1977/tcp	TCO Address Book
tcoaddressbook	1977/udp	TCO Address Book
#	4548-4566	Unassigned
tram	4567/tcp	TRAM
tram	4567/udp	TRAM
bmc-reporting	4568/tcp	BMC Reporting
bmc-reporting	4568/udp	BMC Reporting
#	4569-4599	Unassigned
piranha1	4600/tcp	Piranha1
piranha1	4600/udp	Piranha1
piranha2	4601/tcp	Piranha2
piranha2	4601/udp	Piranha2
#	4602-4671	Unassigned
rfa	4672/tcp	remote file acc
rfa	4672/udp	remote file acc
#	4673-4799	Unassigned
iims	4800/tcp	Icona Instant
iims	4800/udp	Icona Instant
iwec	4801/tcp	Icona Web
iwec	4801/udp	Icona Web
ilss	4802/tcp	Icona
ilss	4802/udp	Icona
#	4803-4826	Unassigned
htcp	4827/tcp	HTCP
htcp	4827/udp	HTCP
#	4828-4836	Unassigned
varadero-0	4837/tcp	Varadero-0
varadero-0	4837/udp	Varadero-0
varadero-1	4838/tcp	Varadero-1
varadero-1	4838/udp	Varadero-1
varadero-2	4839/udp	Varadero-2
varadero-2	4839/udp	Varadero-2
#	4840-4867	Unassigned
phrelay	4868/tcp	Photon Relay
phrelay	4868/udp	Photon Relay
phrelaydbg	4869/tcp	Photon Relay
phrelaydbg	4869/udp	Photon Relay
#	4870-4884	Unassigned
abbs	4885/tcp	ABBS
abbs	4885/udp	ABBS
#	4886-4982	Unassigned
att-intercom	4983/tcp	AT&T Intercom
att-intercom	4983/udp	AT&T Intercom
#	4984-4999	Unassigned
commplex-main	5000/tcp	
commplex-main	5000/udp	
commplex-link	5001/tcp	
commplex-link	5001/udp	
rfe	5002/tcp	radio free eth
rfe	5002/udp	radio free eth
fmpro-internal	5003/tcp	FileMaker, Inc.
fmpro-internal	5003/udp	FileMaker, Inc.
avt-profile-1	5004/tcp	avt-profile-1
avt-profile-1	5004/udp	avt-profile-1
avt-profile-2	5005/tcp	avt-profile-2
avt-profile-2	5005/udp	avt-profile-2
wsm-server	5006/tcp	wsm server
wsm-server	5006/udp	wsm server
wsm-server-ssl	5007/tcp	wsm server ssl
wsm-server-ssl	5007/udp	wsm server ssl
#	5008-5009	Unassigned
telelpathstart	5010/tcp	TelepathStart
telelpathstart	5010/udp	TelepathStart
telelpathattack	5011/tcp	TelepathAttack
telelpathattack	5011/udp	TelepathAttack
#	5012-5019	Unassigned
zenginkyo-1	5020/tcp	zenginkyo-1
zenginkyo-1	5020/udp	zenginkyo-1
zenginkyo-2	5021/tcp	zenginkyo-2
zenginkyo-2	5021/udp	zenginkyo-2
#	5022-5041	Unassigned
asnaacceler8db	5042/tcp	asnaacceler8db
asnaacceler8db	5042/udp	asnaacceler8db

```
unisql          1978/tcp   UniSQL
unisql          1978/udp   UniSQL
unisql-java     1979/tcp   UniSQL Java
unisql-java     1979/udp   UniSQL Java
#               1980-1983  Unassigned
bb              1984/tcp   BB
bb              1984/udp   BB
hsrp            1985/tcp   Hot Standby
hsrp            1985/udp   Hot Standby
licensedaemon   1986/tcp   cisco
licensedaemon   1986/udp   cisco
tr-rsrb-p1      1987/tcp   cisco RSRB
tr-rsrb-p1      1987/udp   cisco RSRB
tr-rsrb-p2      1988/tcp   cisco RSRB
tr-rsrb-p2      1988/udp   cisco RSRB
tr-rsrb-p3      1989/tcp   cisco RSRB
tr-rsrb-p3      1989/udp   cisco RSRB
mshnet          1989/tcp   MHSnet system
mshnet          1989/udp   MHSnet system
stun-p1         1990/tcp   cisco STUN 1
stun-p1         1990/udp   cisco STUN 1
stun-p2         1991/tcp   cisco STUN 2
stun-p2         1991/udp   cisco STUN 2
stun-p3         1992/tcp   cisco STUN 3
stun-p3         1992/udp   cisco STUN 3
ipsendmsg       1992/tcp   IPsendmsg
ipsendmsg       1992/udp   IPsendmsg
snmp-tcp-port   1993/tcp   cisco SNMP TCP
snmp-tcp-port   1993/udp   cisco SNMP TCP
stun-port       1994/tcp   cisco serial
stun-port       1994/udp   cisco serial
perf-port       1995/tcp   cisco perf port
perf-port       1995/udp   cisco perf port
tr-rsrb-port    1996/tcp   cisco Remote SRB
tr-rsrb-port    1996/udp   cisco Remote SRB
gdp-port        1997/tcp   cisco Gateway
gdp-port        1997/udp   cisco Gateway
x25-svc-port    1998/tcp   cisco X.25 (XOT)
x25-svc-port    1998/udp   cisco X.25 (XOT)
tcp-id-port     1999/tcp   cisco ident port
tcp-id-port     1999/udp   cisco ident port
callbook        2000/tcp
callbook        2000/udp
dc              2001/tcp
wizard          2001/udp   curry
globe           2002/tcp
globe           2002/udp
mailbox         2004/tcp
emce            2004/udp    CCWS mm conf
berknet         2005/tcp
oracle          2005/udp
invokator       2006/tcp
raid-cc         2006/udp    raid
dectalk         2007/tcp
raid-am         2007/udp
conf            2008/tcp
terminaldb      2008/udp
news            2009/tcp
whosockami      2009/udp
search          2010/tcp
pipe_server     2010/udp
raid-cc         2011/tcp    raid
servserv        2011/udp
ttyinfo         2012/tcp
raid-ac         2012/udp
raid-am         2013/tcp
raid-cd         2013/udp
troff           2014/tcp
raid-sf         2014/udp
cypress         2015/tcp
raid-cs         2015/udp
#               5043-5049  Unassigned
mmcc            5050/tcp   multimedia
mmcc            5050/udp   multimedia
ita-agent       5051/tcp   ITA Agent
ita-agent       5051/udp   ITA Agent
ita-manager     5052/tcp   ITA Manager
ita-manager     5052/udp   ITA Manager
#               5053-5054  Unassigned
unot            5055/tcp   UNOT
unot            5055/udp   UNOT
#               5056-5059  Unassigned
sip             5060/tcp   SIP
sip             5060/udp   SIP
#               5061-5068  Unassigned
i-net-2000-npr  5069/tcp   I/Net 2000-NPR
i-net-2000-npr  5069/udp   I/Net 2000-NPR
#               5070       Unassigned
powerschool     5071/tcp   PowerSchool
powerschool     5071/udp   PowerSchool
#               5072-5092  Unassigned
sentinel-lm     5093/tcp   Sentinel LM
sentinel-lm     5093/udp   Sentinel LM
#               5094-5098  Unassigned
sentlm-srv2srv  5099/tcp   SentLM Srv2Srv
sentlm-srv2srv  5099/udp   SentLM Srv2Srv
#               5100-5144  Unassigned
rmonitor_secure 5145/tcp   RMONITOR SECURE
rmonitor_secure 5145/udp   RMONITOR SECURE
#               5146-5149  Unassigned
atmp            5150/tcp   Ascend Tunnel
atmp            5150/udp   Ascend Tunnel
esri_sde        5151/tcp   ESRI SDE
esri_sde        5151/udp   ESRI SDE
sde-discovery   5152/tcp   ESRI SDE
sde-discovery   5152/udp   ESRI SDE
#               5153-5164  Unassigned
ife_icorp       5165/tcp   ife_1corp
ife_icorp       5165/udp   ife_1corp
#               5166-5189  Unassigned
aol             5190/tcp   America-Online
aol             5190/udp   America-Online
aol-1           5191/tcp   AmericaOnline1
aol-1           5191/udp   AmericaOnline1
aol-2           5192/tcp   AmericaOnline2
aol-2           5192/udp   AmericaOnline2
aol-3           5193/tcp   AmericaOnline3
aol-3           5193/udp   AmericaOnline3
#               5194-5199  Unassigned
targus-aib1     5200/tcp   Targus AIB 1
targus-aib1     5200/udp   Targus AIB 1
targus-aib2     5201/tcp   Targus AIB 2
targus-aib2     5201/udp   Targus AIB 2
targus-tnts1    5202/tcp   Targus TNTS 1
targus-tnts1    5202/udp   Targus TNTS 1
targus-tnts2    5203/tcp   Targus TNTS 2
targus-tnts2    5203/udp   Targus TNTS 2
#               5204-5235  Unassigned
padl2sim        5236/tcp
padl2sim        5236/udp
#               5237-5271  Unassigned
pk              5272/tcp   PK
pk              5272/udp   PK
#               5273-5299  Unassigned
hacl-hb         5300/tcp
hacl-hb         5300/udp
hacl-gs         5301/tcp
hacl-gs         5301/udp
hacl-cfg        5302/tcp
hacl-cfg        5302/udp
hacl-probe      5303/tcp
hacl-probe      5303/udp
```

bootserver	2016/tcp	
bootserver	2016/udp	
cypress-stat	2017/tcp	
bootclient	2017/udp	
terminaldb	2018/tcp	
rellpack	2018/udp	
whosockami	2019/tcp	
about	2019/udp	
xinupageserver	2020/tcp	
xinupageserver	2020/udp	
servexec	2021/tcp	
xinuexpansion1	2021/udp	
down	2022/tcp	
xinuexpansion2	2022/udp	
xinuexpansion3	2023/tcp	
xinuexpansion3	2023/udp	
xinuexpansion4	2024/tcp	
xinuexpansion4	2024/udp	
ellpack	2025/tcp	
xribs	2025/udp	
scrabble	2026/tcp	
scrabble	2026/udp	
shadowserver	2027/tcp	
shadowserver	2027/udp	
submitserver	2028/tcp	
submitserver	2028/udp	
device2	2030/tcp	
device2	2030/udp	
blackboard	2032/tcp	
blackboard	2032/udp	
glogger	2033/tcp	
glogger	2033/udp	
scoremgr	2034/tcp	
scoremgr	2034/udp	
imsldoc	2035/tcp	
imsldoc	2035/udp	
objectmanager	2038/tcp	
objectmanager	2038/udp	
lam	2040/tcp	
lam	2040/udp	
interbase	2041/tcp	
interbase	2041/udp	
isis	2042/tcp	isis
isis	2042/udp	isis
isis-bcast	2043/tcp	isis-bcast
isis-bcast	2043/udp	isis-bcast
rimsl	2044/tcp	
rimsl	2044/udp	
cdfunc	2045/tcp	
cdfunc	2045/udp	
sdfunc	2046/tcp	
sdfunc	2046/udp	
dls	2047/tcp	
dls	2047/udp	
dls-monitor	2048/tcp	
dls-monitor	2048/udp	
shilp	2049/tcp	
shilp	2049/udp	
nfs	2049/tcp	Network File Sys
nfs	2049/udp	Network File Sys
dlsrpn	2065/tcp	Data Link Switch
dlsrpn	2065/udp	Data Link Switch
dlswpn	2067/tcp	Data Link Switch
dlswpn	2067/udp	Data Link Switch
lrp	2090/tcp	Load Report
lrp	2090/udp	Load Report
prp	2091/tcp	PRP
prp	2091/udp	PRP
descent3	2092/tcp	Descent 3
descent3	2092/udp	Descent 3
nbx-cc	2093/tcp	NBX CC
hacl-local	5304/tcp	
hacl-local	5304/udp	
hacl-test	5305/tcp	
hacl-test	5305/udp	
sun-mc-grp	5306/tcp	Sun MC Group
sun-mc-grp	5306/udp	Sun MC Group
sco-aip	5307/tcp	SCO AIP
sco-aip	5307/udp	SCO AIP
cfengine	5308/tcp	CFengine
cfengine	5308/udp	CFengine
jprinter	5309/tcp	J Printer
jprinter	5309/udp	J Printer
outlaws	5310/tcp	Outlaws
outlaws	5310/udp	Outlaws
tmlogin	5311/tcp	TM Login
tmlogin	5311/udp	TM Login
#	5312-5399	Unassigned
excerpt	5400/tcp	Excerpt Search
excerpt	5400/udp	Excerpt Search
excerpts	5401/tcp	Excerpt Search
excerpts	5401/udp	Excerpt Search
mftp	5402/tcp	MFTP
mftp	5402/udp	MFTP
hpoms-ci-lstn	5403/tcp	HPOMS-CI-LSTN
hpoms-ci-lstn	5403/udp	HPOMS-CI-LSTN
hpoms-dps-lstn	5404/tcp	HPOMS-DPS-LSTN
hpoms-dps-lstn	5404/udp	HPOMS-DPS-LSTN
netsupport	5405/tcp	NetSupport
netsupport	5405/udp	NetSupport
systemics-sox	5406/tcp	Systemics Sox
systemics-sox	5406/udp	Systemics Sox
foresyte-clear	5407/tcp	Foresyte-Clear
foresyte-clear	5407/udp	Foresyte-Clear
foresyte-sec	5408/tcp	Foresyte-Sec
foresyte-sec	5408/udp	Foresyte-Sec
salient-dtasrv	5409/tcp	Salient Data
salient-dtasrv	5409/udp	Salient Data
salient-usrmgr	5410/tcp	Salient User Mgr
salient-usrmgr	5410/udp	Salient User Mgr
actnet	5411/tcp	ActNet
actnet	5411/udp	ActNet
continuus	5412/tcp	Continuus
continuus	5412/udp	Continuus
wwiotalk	5413/tcp	WWIOTALK
wwiotalk	5413/udp	WWIOTALK
statusd	5414/tcp	StatusD
statusd	5414/udp	StatusD
ns-server	5415/tcp	NS Server
ns-server	5415/udp	NS Server
sns-gateway	5416/tcp	SNS Gateway
sns-gateway	5416/udp	SNS Gateway
sns-agent	5417/tcp	SNS Agent
sns-agent	5417/udp	SNS Agent
mcntp	5418/tcp	MCNTP
mcntp	5418/udp	MCNTP
dj-ice	5419/tcp	DJ-ICE
dj-ice	5419/udp	DJ-ICE
cylink-c	5420/tcp	Cylink-C
cylink-c	5420/udp	Cylink-C
netsupport2	5421/tcp	Net Support 2
netsupport2	5421/udp	Net Support 2
salient-mux	5422/tcp	Salient MUX
salient-mux	5422/udp	Salient MUX
virtualuser	5423/tcp	VIRTUALUSER
virtualuser	5423/udp	VIRTUALUSER
#	5424-5425	Unassigned
devbasic	5426/tcp	DEVBASIC
devbasic	5426/udp	DEVBASIC
sco-peer-tta	5427/tcp	SCO-PEER-TTA
sco-peer-tta	5427/udp	SCO-PEER-TTA
telaconsole	5428/tcp	TELACONSOLE

nbx-cc	2093/udp	NBX CC
nbx-au	2094/tcp	NBX AU
nbx-au	2094/udp	NBX AU
nbx-ser	2095/tcp	NBX SER
nbx-ser	2095/udp	NBX SER
nbx-dir	2096/tcp	NBX DIR
nbx-dir	2096/udp	NBX DIR
jetformpreview	2097/tcp	Jet Form Preview
jetformpreview	2097/udp	Jet Form Preview
dialog-port	2098/tcp	Dialog Port
dialog-port	2098/udp	Dialog Port
h2250-annex-g	2099/tcp	H.225.0 Annex G
h2250-annex-g	2099/udp	H.225.0 Annex G
amiganetfs	2100/tcp	amiganetfs
amiganetfs	2100/udp	amiganetfs
rtcm-sc104	2101/tcp	rtcm-sc104
rtcm-sc104	2101/udp	rtcm-sc104
zephyr-srv	2102/tcp	Zephyr server
zephyr-srv	2102/udp	Zephyr server
zephyr-clt	2103/tcp	Zephyr serv-hm
zephyr-clt	2103/udp	Zephyr serv-hm
zephyr-hm	2104/tcp	Zephyr hostman
zephyr-hm	2104/udp	Zephyr hostman
minipay	2105/tcp	MiniPay
minipay	2105/udp	MiniPay
mzap	2106/tcp	MZAP
mzap	2106/udp	MZAP
bintec-admin	2107/tcp	BinTec Admin
bintec-admin	2107/udp	BinTec Admin
comcam	2108/tcp	Comcam
comcam	2108/udp	Comcam
ergolight	2109/tcp	Ergolight
ergolight	2109/udp	Ergolight
umsp	2110/tcp	UMSP
umsp	2110/udp	UMSP
dsatp	2111/tcp	DSATP
dsatp	2111/udp	DSATP
idonix-metanet	2112/tcp	Idonix MetaNet
idonix-metanet	2112/udp	Idonix MetaNet
hsl-storm	2113/tcp	HSL StoRM
hsl-storm	2113/udp	HSL StoRM
newheights	2114/tcp	NEWHEIGHTS
newheights	2114/udp	NEWHEIGHTS
kdm	2115/tcp	KDM
kdm	2115/udp	KDM
ccowcmr	2116/tcp	CCOWCMR
ccowcmr	2116/udp	CCOWCMR
mentaclient	2117/tcp	MENTACLIENT
mentaclient	2117/udp	MENTACLIENT
mentaserver	2118/tcp	MENTASERVER
mentaserver	2118/udp	MENTASERVER
gsigatekeeper	2119/tcp	GSIGATEKEEPER
gsigatekeeper	2119/udp	GSIGATEKEEPER
qencp	2120/tcp	Quick Eagle CP
qencp	2120/udp	Quick Eagle CP
scientia-ssdb	2121/tcp	SCIENTIA-SSDB
scientia-ssdb	2121/udp	SCIENTIA-SSDB
caupc-remote	2122/tcp	CauPC Remote Ctl
caupc-remote	2122/udp	CauPC Remote Ctl
gtp-control	2123/tcp	GTP-Control 3GPP
gtp-control	2123/udp	GTP-Control 3GPP
elatelink	2124/tcp	ELATELINK
elatelink	2124/udp	ELATELINK
lockstep	2125/tcp	LOCKSTEP
lockstep	2125/udp	LOCKSTEP
pktcable-cops	2126/tcp	PktCable-COPS
pktcable-cops	2126/udp	PktCable-COPS
index-pc-wb	2127/tcp	INDEX-PC-WB
index-pc-wb	2127/udp	INDEX-PC-WB
net-steward	2128/tcp	Net Steward Ctl
net-steward	2128/udp	Net Steward Ctl
telaconsole	5428/udp	TELACONSOLE
base	5429/tcp	Billing and Acc
base	5429/udp	Billing and Acc
radec-corp	5430/tcp	RADEC CORP
radec-corp	5430/udp	RADEC CORP
park-agent	5431/tcp	PARK AGENT
park-agnet	5431/udp	PARK AGENT
#	5432-5434	Unassigned
dttl	5435/tcp	Data (DTTL)
dttl	5435/udp	Data (DTTL)
#	5436-5453	Unassigned
apc-tcp-udp-4	5454/tcp	apc-tcp-udp-4
apc-tcp-udp-4	5454/udp	apc-tcp-udp-4
apc-tcp-udp-5	5455/tcp	apc-tcp-udp-5
apc-tcp-udp-5	5455/udp	apc-tcp-udp-5
apc-tcp-udp-6	5456/tcp	apc-tcp-udp-6
apc-tcp-udp-6	5456/udp	apc-tcp-udp-6
#	5457-5460	Unassigned
silkmeter	5461/tcp	SILKMETER
silkmeter	5461/udp	SILKMETER
ttl-publisher	5462/tcp	TTL Publisher
ttl-publisher	5462/udp	TTL Publisher
#	5463-5464	Unassigned
netops-broker	5465/tcp	NETOPS-BROKER
netops-broker	5465/udp	NETOPS-BROKER
#	5466-5499	Unassigned
fcp-addr-srvr1	5500/tcp	fcp-addr-srvr1
fcp-addr-srvr1	5500/udp	fcp-addr-srvr1
fcp-addr-srvr2	5501/tcp	fcp-addr-srvr2
fcp-addr-srvr2	5501/udp	fcp-addr-srvr2
fcp-srvr-inst1	5502/tcp	fcp-srvr-inst1
fcp-srvr-inst1	5502/udp	fcp-srvr-inst1
fcp-srvr-inst2	5503/tcp	fcp-srvr-inst2
fcp-srvr-inst2	5503/udp	fcp-srvr-inst2
fcp-cics-gw1	5504/tcp	fcp-cics-gw1
fcp-cics-gw1	5504/udp	fcp-cics-gw1
#	5504-5553	Unassigned
sgi-esphttp	5554/tcp	SGI ESP HTTP
sgi-esphttp	5554/udp	SGI ESP HTTP
personal-agent	5555/tcp	Personal Agent
personal-agent	5555/udp	Personal Agent
#	5556-5598	Unassigned
esinstall	5599/tcp	Enterprise
esinstall	5599/udp	Enterprise
esmmanager	5600/tcp	Enterprise
esmmanager	5600/udp	Enterprise
esmagent	5601/tcp	Enterprise
esmagent	5601/udp	Enterprise
a1-msc	5602/tcp	A1-MSC
a1-msc	5602/udp	A1-MSC
a1-bs	5603/tcp	A1-BS
a1-bs	5603/udp	A1-BS
a3-sdunode	5604/tcp	A3-SDUNode
a3-sdunode	5604/udp	A3-SDUNode
a4-sdunode	5605/tcp	A4-SDUNode
a4-sdunode	5605/udp	A4-SDUNode
#	5606-5630	Unassigned
pcanywheredata	5631/tcp	pcANYWHEREdata
pcanywheredata	5631/udp	pcANYWHEREdata
pcanywherestat	5632/tcp	pcANYWHEREstat
pcanywherestat	5632/udp	pcANYWHEREstat
#	5633-5677	Unassigned
rrac	5678/tcp	Remote RAC
rrac	5678/udp	Remote RAC
dccm	5679/tcp	Direct Cable Mgr
dccm	5679/udp	Direct Cable Mgr
#	5780-5712	Unassigned
proshareaudio	5713/tcp	proshare audio
proshareaudio	5713/udp	proshare audio
prosharevideo	5714/tcp	proshare video
prosharevideo	5714/udp	proshare video

cs-live	2129/tcp	cs-live.com
cs-live	2129/udp	cs-live.com
swc-xds	2130/tcp	SWC-XDS
swc-xds	2130/udp	SWC-XDS
avantageb2b	2131/tcp	Avantageb2b
avantageb2b	2131/udp	Avantageb2b
avail-epmap	2132/tcp	AVAIL-EPMAP
avail-epmap	2132/udp	AVAIL-EPMAP
zymed-zpp	2133/tcp	ZYMED-ZPP
zymed-zpp	2133/udp	ZYMED-ZPP
avenue	2134/tcp	AVENUE
avenue	2134/udp	AVENUE
gris	2135/tcp	Grid Resource
gris	2135/udp	Grid Resource
appworxsrv	2136/tcp	APPWORXSRV
appworxsrv	2136/udp	APPWORXSRV
connect	2137/tcp	CONNECT
connect	2137/udp	CONNECT
unbind-cluster	2138/tcp	UNBIND-CLUSTER
unbind-cluster	2138/udp	UNBIND-CLUSTER
ias-auth	2139/tcp	IAS-AUTH
ias-auth	2139/udp	IAS-AUTH
ias-reg	2140/tcp	IAS-REG
ias-reg	2140/udp	IAS-REG
ias-admind	2141/tcp	IAS-ADMIND
ias-admind	2141/udp	IAS-ADMIND
tdm-over-ip	2142/tcp	TDM-OVER-IP
tdm-over-ip	2142/udp	TDM-OVER-IP
lv-jc	2143/tcp	Live Vault
lv-jc	2143/udp	Live Vault
lv-ffx	2144/tcp	Live Vault
lv-ffx	2144/udp	Live Vault
lv-pici	2145/tcp	Live Vault
lv-pici	2145/udp	Live Vault
lv-not	2146/tcp	Live Vault
lv-not	2146/udp	Live Vault
lv-auth	2147/tcp	Live Vault
lv-auth	2147/udp	Live Vault
veritas-ucl	2148/tcp	VERITAS
veritas-ucl	2148/udp	VERITAS
acptsys	2149/tcp	ACPTSYS
acptsys	2149/udp	ACPTSYS
dynamic3d	2150/tcp	DYNAMIC3D
dynamic3d	2150/udp	DYNAMIC3D
docent	2151/tcp	DOCENT
docent	2151/udp	DOCENT
gtp-user	2152/tcp	GTP-User (3GPP)
gtp-user	2152/udp	GTP-User (3GPP)
#	2153-2164	Unassigned
x-bone-api	2165/tcp	X-Bone API
x-bone-api	2165/udp	X-Bone API
iwserver	2166/tcp	IWSERVER
iwserver	2166/udp	IWSERVER
#	2167-2179	Unassigned
mc-gt-srv	2180/tcp	MVGS
mc-gt-srv	2180/udp	MVGS
eforward	2181/tcp	eforward
eforward	2181/udp	eforward
ici	2200/tcp	ICI
ici	2200/udp	ICI
ats	2201/tcp	ATSP
ats	2201/udp	ATSP
imtc-map	2202/tcp	Int. Multimedia
imtc-map	2202/udp	Int. Multimedia
kali	2213/tcp	Kali
kali	2213/udp	Kali
ganymede	2220/tcp	Ganymede
ganymede	2220/udp	Ganymede
rockwell-csp1	2221/tcp	Rockwell CSP1
rockwell-csp1	2221/udp	Rockwell CSP1
rockwell-csp2	2222/tcp	Rockwell CSP2
prosharedata	5715/tcp	proshare data
prosharedata	5715/udp	proshare data
prosharerequest	5716/tcp	proshare request
prosharerequest	5716/udp	proshare request
prosharenotify	5717/tcp	proshare notify
prosharenotify	5717/udp	proshare notify
#	5718-5728	Unassigned
openmail	5729/tcp	Openmail
openmail	5729/udp	Openmail
#	5730-5740	Unassigned
ida-discover1	5741/tcp	IDA Disc Port 1
ida-discover1	5741/udp	IDA Disc Port 1
ida-discover2	5742/tcp	IDA Disc Port 2
ida-discover2	5742/udp	IDA Disc Port 2
#	5743-5744	Unassigned
fcopy-server	5745/tcp	fcopy-server
fcopy-server	5745/udp	fcopy-server
fcopys-server	5746/tcp	fcopys-server
fcopys-server	5746/udp	fcopys-server
#	5769-5770	Unassigned
netagent	5771/tcp	NetAgent
netagent	5771/udp	NetAgent
#	5772-5812	Unassigned
icmpd	5813/tcp	ICMPD
icmpd	5813/udp	ICMPD
#	5814-5858	Unassigned
wherehoo	5859/tcp	WHEREHOO
wherehoo	5859/udp	WHEREHOO
#	5860-5967	Unassigned
mppolicy-v5	5968/tcp	mppolicy-v5
mppolicy-v5	5968/udp	mppolicy-v5
mppolicy-mgr	5969/tcp	mppolicy-mgr
mppolicy-mgr	5969/udp	mppolicy-mgr
#	5970-5998	Unassigned
cvsup	5999/tcp	CVSup
cvsup	5999/udp	CVSup
x11	6000-6063/tcp	X Window
x11	6000-6063/udp	X Window
ndl-ahp-svc	6064/tcp	NDL-AHP-SVC
ndl-ahp-svc	6064/udp	NDL-AHP-SVC
winpharaoh	6065/tcp	WinPharaoh
winpharaoh	6065/udp	WinPharaoh
ewctsp	6066/tcp	EWCTSP
ewctsp	6066/udp	EWCTSP
srb	6067/tcp	SRB
srb	6067/udp	SRB
gsmp	6068/tcp	GSMP
gsmp	6068/udp	GSMP
trip	6069/tcp	TRIP
trip	6069/udp	TRIP
messageasap	6070/tcp	Messageasap
messageasap	6070/udp	Messageasap
ssdtp	6071/tcp	SSDTP
ssdtp	6071/udp	SSDTP
diagnose-proc	6072/tcp	DIAGNOSE-PROC
diagmose-proc	6072/udp	DIAGNOSE-PROC
directplay8	6073/tcp	DirectPlay8
directplay8	6073/udp	DirectPlay8
#	6074-6099	Unassigned
synchronet-db	6100/tcp	SynchroNet-db
synchronet-db	6100/udp	SynchroNet-db
synchronet-rtc	6101/tcp	SynchroNet-rtc
synchronet-rtc	6101/udp	SynchroNet-rtc
synchronet-upd	6102/tcp	SynchroNet-upd
synchronet-upd	6102/udp	SynchroNet-upd
rets	6103/tcp	RETS
rets	6103/udp	RETS
dbdb	6104/tcp	DBDB
dbdb	6104/udp	DBDB
primaserver	6105/tcp	Prima Server
primaserver	6105/udp	Prima Server

rockwell-csp2	2222/udp	Rockwell CSP2
rockwell-csp3	2223/tcp	Rockwell CSP3
rockwell-csp3	2223/udp	Rockwell CSP3
ivs-video	2232/tcp	IVS Video
ivs-video	2232/udp	IVS Video
infocrypt	2233/tcp	INFOCRYPT
infocrypt	2233/udp	INFOCRYPT
directplay	2234/tcp	DirectPlay
directplay	2234/udp	DirectPlay
sercomm-wlink	2235/tcp	Sercomm-WLink
sercomm-wlink	2235/udp	Sercomm-WLink
nani	2236/tcp	Nani
nani	2236/udp	Nani
optech-port1-lm	2237/tcp	Optech Port1
optech-port1-lm	2237/udp	Optech Port1
aviva-sna	2238/tcp	AVIVA SNA SERVER
aviva-sna	2238/udp	AVIVA SNA SERVER
imagequery	2239/tcp	Image Query
imagequery	2239/udp	Image Query
recipe	2240/tcp	RECIPe
recipe	2240/udp	RECIPe
ivsd	2241/tcp	IVS Daemon
ivsd	2241/udp	IVS Daemon
foliocorp	2242/tcp	Folio Remote
foliocorp	2242/udp	Folio Remote
magicom	2243/tcp	Magicom Protocol
magicom	2243/udp	Magicom Protocol
nmsserver	2244/tcp	NMS Server
nmsserver	2244/udp	NMS Server
hao	2245/tcp	HaO
hao	2245/udp	HaO
#	2245-2278	Unassigned
xmquery	2279/tcp	xmquery
xmquery	2279/udp	xmquery
lnvpoller	2280/tcp	LNVPOLLER
lnvpoller	2280/udp	LNVPOLLER
lnvconsole	2281/tcp	LNVCONSOLE
lnvconsole	2281/udp	LNVCONSOLE
lnvalarm	2282/tcp	LNVALARM
lnvalarm	2282/udp	LNVALARM
lnvstatus	2283/tcp	LNVSTATUS
lnvstatus	2283/udp	LNVSTATUS
lnvmaps	2284/tcp	LNVMAPS
lnvmaps	2284/udp	LNVMAPS
lnvmailmon	2285/tcp	LNVMAILMON
lnvmailmon	2285/udp	LNVMAILMON
nas-metering	2286/tcp	NAS-Metering
nas-metering	2286/udp	NAS-Metering
dna	2287/tcp	DNA
dna	2287/udp	DNA
netml	2288/tcp	NETML
netml	2288/udp	NETML
#	2289-2293	Unassigned
konshus-lm	2294/tcp	Konshus (FLEX)
konshus-lm	2294/udp	Konshus (FLEX)
advant-lm	2295/tcp	Advant
advant-lm	2295/udp	Advant
theta-lm	2296/tcp	Theta (Rainbow)
theta-lm	2296/udp	Theta (Rainbow)
d2k-datamover1	2297/tcp	D2K DataMover 1
d2k-datamover1	2297/udp	D2K DataMover 1
d2k-datamover2	2298/tcp	D2K DataMover 2
d2k-datamover2	2298/udp	D2K DataMover 2
pc-telecommute	2299/tcp	PC Telecommute
pc-telecommute	2299/udp	PC Telecommute
cvmmon	2300/tcp	CVMMON
cvmmon	2300/udp	CVMMON
cpq-wbem	2301/tcp	Compaq HTTP
cpq-wbem	2301/udp	Compaq HTTP
binderysupport	2302/tcp	Bindery Support
binderysupport	2302/udp	Bindery Support
mpsserver	6106/tcp	MPS Server
mpsserver	6106/udp	MPS Server
etc-control	6107/tcp	ETC Control
etc-control	6107/udp	ETC Control
sercomm-scadmin	6108/tcp	Sercomm-SCAdmin
sercomm-scadmin	6108/udp	Sercomm-SCAdmin
globecast-id	6109/tcp	GLOBECAST-ID
globecast-id	6109/udp	GLOBECAST-ID
softcm	6110/tcp	HP SoftBench CM
softcm	6110/udp	HP SoftBench CM
spc	6111/tcp	HP SoftBench
spc	6111/udp	HP SoftBench
dtspcd	6112/tcp	dtspcd
dtspcd	6112/udp	dtspcd
#	6113-6122	Unassigned
backup-express	6123/tcp	Backup Express
backup-express	6123/udp	Backup Express
#	6124-6140	Unassigned
meta-corp	6141/tcp	Meta Corporation
meta-corp	6141/udp	Meta Corporation
aspentec-lm	6142/tcp	Aspen Technology
aspentec-lm	6142/udp	Aspen Technology
watershed-lm	6143/tcp	Watershed
watershed-lm	6143/udp	Watershed
statsci1-lm	6144/tcp	StatSci - 1
statsci1-lm	6144/udp	StatSci - 1
statsci2-lm	6145/tcp	StatSci - 2
statsci2-lm	6145/udp	StatSci - 2
lonewolf-lm	6146/tcp	Lone Wolf
lonewolf-lm	6146/udp	Lone Wolf
montage-lm	6147/tcp	Montage
montage-lm	6147/udp	Montage
ricardo-lm	6148/tcp	Ricardo America
ricardo-lm	6148/udp	Ricardo America
tal-pod	6149/tcp	tal-pod
tal-pod	6149/udp	tal-pod
#	6150-6252	Unassigned
crip	6253/tcp	CRIP
crip	6253/udp	CRIP
#	6254-6320	Unassigned
emp-server1	6321/tcp	Empress Software
emp-server1	6321/udp	Empress Software
emp-server2	6322/tcp	Empress Software
emp-server2	6322/udp	Empress Software
#	6323-6388	Unassigned
clariion-evr01	6389/tcp	clariion-evr01
clariion-evr01	6389/udp	clariion-evr01
#	6390-6399	Unassigned
info-aps	6400	
info-was	6401	
info-eventsvr	6402	
info-cachesvr	6403	
info-filesvr	6404	
info-pagesvr	6405	
info-processvr	6406	
reserved1	6407	
reserved2	6408	
reserved3	6409	
reserved4	6410	
#	6411-6454	Unassigned
skip-cert-recv	6455/tcp	SKIP Certificate
skip-cert-send	6456/tcp	SKIP Certificate
#	6457-6470	Unassigned
lvision-lm	6471/tcp	LVision
lvision-lm	6471/udp	LVision
#	6472-6499	Unassigned
boks	6500/tcp	BoKS Master
boks	6500/udp	BoKS Master
boks_servc	6501/tcp	BoKS Servc
boks_servc	6501/udp	BoKS Servc
boks_servm	6502/tcp	BoKS Servm

```
proxy-gateway   2303/tcp   Proxy Gateway
proxy-gateway   2303/udp   Proxy Gateway
attachmate-uts  2304/tcp   Attachmate UTS
attachmate-uts  2304/udp   Attachmate UTS
mt-scaleserver  2305/tcp   MT ScaleServer
mt-scaleserver  2305/udp   MT ScaleServer
tappi-boxnet    2306/tcp   TAPPI BoxNet
tappi-boxnet    2306/udp   TAPPI BoxNet
pehelp          2307/tcp   pehelp
pehelp          2307/udp   pehelp
sdhelp          2308/tcp   sdhelp
sdhelp          2308/udp   sdhelp
sdserver        2309/tcp   SD Server
sdserver        2309/udp   SD Server
sdclient        2310/tcp   SD Client
sdclient        2310/udp   SD Client
messageservice  2311/tcp   Message Service
messageservice  2311/udp   Message Service
iapp            2313/tcp   IAPP
iapp            2313/udp   IAPP
cr-websystems   2314/tcp   CR WebSystems
cr-websystems   2314/udp   CR WebSystems
precise-sft     2315/tcp   Precise Sft.
precise-sft     2315/udp   Precise Sft.
sent-lm         2316/tcp   SENT
sent-lm         2316/udp   SENT
attachmate-g32  2317/tcp   Attachmate G32
attachmate-g32  2317/udp   Attachmate G32
cadencecontrol  2318/tcp   Cadence Control
cadencecontrol  2318/udp   Cadence Control
infolibria      2319/tcp   InfoLibria
infolibria      2319/udp   InfoLibria
siebel-ns       2320/tcp   Siebel NS
siebel-ns       2320/udp   Siebel NS
rdlap           2321/tcp   RDLAP over UDP
rdlap           2321/udp   RDLAP
ofsd            2322/tcp   ofsd
ofsd            2322/udp   ofsd
3d-nfsd         2323/tcp   3d-nfsd
3d-nfsd         2323/udp   3d-nfsd
cosmocall       2324/tcp   Cosmocall
cosmocall       2324/udp   Cosmocall
designspace-lm  2325/tcp   Design Space
designspace-lm  2325/udp   Design Space
idcp            2326/tcp   IDCP
idcp            2326/udp   IDCP
xingcsm         2327/tcp   xingcsm
xingcsm         2327/udp   xingcsm
netrix-sftm     2328/tcp   Netrix SFTM
netrix-sftm     2328/udp   Netrix SFTM
nvd             2329/tcp   NVD
nvd             2329/udp   NVD
tscchat         2330/tcp   TSCCHAT
tscchat         2330/udp   TSCCHAT
agentview       2331/tcp   AGENTVIEW
agentview       2331/udp   AGENTVIEW
rcc-host        2332/tcp   RCC Host
rcc-host        2332/udp   RCC Host
snapp           2333/tcp   SNAPP
snapp           2333/udp   SNAPP
ace-client      2334/tcp   ACE Client Auth
ace-client      2334/udp   ACE Client Auth
ace-proxy       2335/tcp   ACE Proxy
ace-proxy       2335/udp   ACE Proxy
appleugcontrol  2336/tcp   Apple UG Control
appleugcontrol  2336/udp   Apple UG Control
ideesrv         2337/tcp   ideesrv
ideesrv         2337/udp   ideesrv
norton-lambert  2338/tcp   Norton Lambert
norton-lambert  2338/udp   Norton Lambert
3com-webview    2339/tcp   3Com WebView
boks_servm      6502/udp   BoKS Servm
boks_clntd      6503/tcp   BoKS Clntd
boks_clntd      6503/udp   BoKS Clntd
#               6504       Unassigned
badm_priv       6505/tcp   BoKS Admin
badm_priv       6505/udp   BoKS Admin
badm_pub        6506/tcp   BoKS Admin
badm_pub        6506/udp   BoKS Admin
bdir_priv       6507/tcp   BoKS Dir Server
bdir_priv       6507/udp   BoKS Dir Server
bdir_pub        6508/tcp   BoKS Dir Server
bdir_pub        6508/udp   BoKS Dir Server
#               6509-6546  Unassigned
apc-tcp-udp-1   6547/tcp   apc-tcp-udp-1
apc-tcp-udp-1   6547/udp   apc-tcp-udp-1
apc-tcp-udp-2   6548/tcp   apc-tcp-udp-2
apc-tcp-udp-2   6548/udp   apc-tcp-udp-2
apc-tcp-udp-3   6549/tcp   apc-tcp-udp-3
apc-tcp-udp-3   6549/udp   apc-tcp-udp-3
fg-sysupdate    6550/tcp   fg-sysupdate
fg-sysupdate    6550/udp   fg-sysupdate
#               6551-6557  Unassigned
xdsxdm          6558/tcp
xdsxdm          6558/udp
ircu            6665-6669/tcp  IRCU
ircu            6665-6669/udp  IRCU
vocaltec-gold   6670/tcp   Vocaltec Global
vocaltec-gold   6670/udp   Vocaltec Global
vision_server   6672/tcp   vision_server
vision_server   6672/udp   vision_server
vision_elmd     6673/tcp   vision_elmd
vision_elmd     6673/udp   vision_elmd
kti-icad-srvr   6701/tcp   KTI/ICAD NS
kti-icad-srvr   6701/udp   KTI/ICAD NS
#               6702-6766  Unassigned
bmc-perf-agent  6767/tcp   BMC PERFORM
bmc-perf-agent  6767/udp   BMC PERFORM
bmc-perf-mgrd   6768/tcp   BMC PERFORM
bmc-perf-mgrd   6768/udp   BMC PERFORM
#               6769-6789  Unassigned
hnmp            6790/tcp   HNMP
hnmp            6790/udp   HNMP
ambit-lm        6831/tcp   ambit-lm
ambit-lm        6831/udp   ambit-lm
netmo-default   6841/tcp   Netmo Default
netmo-default   6841/udp   Netmo Default
netmo-http      6842/tcp   Netmo HTTP
netmo-http      6842/udp   Netmo HTTP
#               6843-6849  Unassigned
iccrushmore     6850/tcp   ICCRUSHMORE
iccrushmore     6850/udp   ICCRUSHMORE
#               6851-6887  Unassigned
muse            6888/tcp   MUSE
muse            6888/udp   MUSE
#               6889-6960  Unassigned
jmact3          6961/tcp   JMACT3
jmact3          6961/udp   JMACT3
jmevt2          6962/tcp   jmevt2
jmevt2          6962/udp   jmevt2
swismgr1        6963/tcp   swismgr1
swismgr1        6963/udp   swismgr1
swismgr2        6964/tcp   swismgr2
swismgr2        6964/udp   swismgr2
swistrap        6965/tcp   swistrap
swistrap        6965/udp   swistrap
swispol         6966/tcp   swispol
swispol         6966/udp   swispol
acmsoda         6969/tcp   acmsoda
acmsoda         6969/udp   acmsoda
iatp-highpri    6998/tcp   IATP-highPri
iatp-highpri    6998/udp   IATP-highPri
```

3com-webview	2339/udp	3Com WebView
wrs_registry	2340/tcp	WRS Registry
wrs_registry	2340/udp	WRS Registry
xiostatus	2341/tcp	XIO Status
xiostatus	2341/udp	XIO Status
manage-exec	2342/tcp	Seagate Manage
manage-exec	2342/udp	Seagate Manage
nati-logos	2343/tcp	nati logos
nati-logos	2343/udp	nati logos
fcmsys	2344/tcp	fcmsys
fcmsys	2344/udp	fcmsys
dbm	2345/tcp	dbm
dbm	2345/udp	dbm
redstorm_join	2346/tcp	Game Connection
redstorm_join	2346/udp	Game Connection
redstorm_find	2347/tcp	Game
redstorm_find	2347/udp	Game
redstorm_info	2348/tcp	Game status
redstorm_info	2348/udp	Game status
redstorm_diag	2349/tcp	Diagnostics Port
redstorm_diag	2349/udp	Disgnostics Port
psbserver	2350/tcp	psbserver
psbserver	2350/udp	psbserver
psrserver	2351/tcp	psrserver
psrserver	2351/udp	psrserver
pslserver	2352/tcp	pslserver
pslserver	2352/udp	pslserver
pspserver	2353/tcp	pspserver
pspserver	2353/udp	pspserver
psprserver	2354/tcp	psprserver
psprserver	2354/udp	psprserver
psdbserver	2355/tcp	psdbserver
psdbserver	2355/udp	psdbserver
gxtelmd	2356/tcp	GXT License Man
gxtelmd	2356/udp	GXT License Man
unihub-server	2357/tcp	UniHub Server
unihub-server	2357/udp	UniHub Server
futrix	2358/tcp	Futrix
futrix	2358/udp	Futrix
flukeserver	2359/tcp	FlukeServer
flukeserver	2359/udp	FlukeServer
nexstorindltd	2360/tcp	NexstorIndLtd
nexstorindltd	2360/udp	NexstorIndLtd
tl1	2361/tcp	TL1
tl1	2361/udp	TL1
digiman	2362/tcp	digiman
digiman	2362/udp	digiman
mediacntrlnfsd	2363/tcp	Media Cent NFSD
mediacntrlnfsd	2363/udp	Media Cent NFSD
oi-2000	2364/tcp	OI-2000
oi-2000	2364/udp	OI-2000
dbref	2365/tcp	dbref
dbref	2365/udp	dbref
qip-login	2366/tcp	qip-login
qip-login	2366/udp	qip-login
service-ctrl	2367/tcp	Service Control
service-ctrl	2367/udp	Service Control
opentable	2368/tcp	OpenTable
opentable	2368/udp	OpenTable
acs2000-dsp	2369/tcp	ACS2000 DSP
acs2000-dsp	2369/udp	ACS2000 DSP
l3-hbmon	2370/tcp	L3-HBMon
l3-hbmon	2370/udp	L3-HBMon
#	2371-2380	Unassigned
compaq-https	2381/tcp	Compaq HTTPS
compaq-https	2381/udp	Compaq HTTPS
ms-olap3	2382/tcp	Microsoft OLAP
ms-olap3	2382/udp	Microsoft OLAP
ms-olap4	2383/tcp	Microsoft OLAP
ms-olap4	2383/udp	Microsoft OLAP
sd-request	2384/tcp	SD-REQUEST
iatp-normalpri	6999/tcp	IATP-normalPri
iatp-normalpri	6999/udp	IATP-normalPri
afs3-fileserver	7000/tcp	file server
afs3-fileserver	7000/udp	file server
afs3-callback	7001/tcp	callbacks
afs3-callback	7001/udp	callbacks
afs3-prserver	7002/tcp	users & groups
afs3-prserver	7002/udp	users & groups
afs3-vlserver	7003/tcp	volume location
afs3-vlserver	7003/udp	volume location
afs3-kaserver	7004/tcp	AFS/Kerberos
afs3-kaserver	7004/udp	AFS/Kerberos
afs3-volser	7005/tcp	volume managment
afs3-volser	7005/udp	volume managment
afs3-errors	7006/tcp	error service
afs3-errors	7006/udp	error service
afs3-bos	7007/tcp	basic overseer
afs3-bos	7007/udp	basic overseer
afs3-update	7008/tcp	server-to-server
afs3-update	7008/udp	server-to-server
afs3-rmtsys	7009/tcp	remote cache
afs3-rmtsys	7009/udp	remote cache
ups-onlinet	7010/tcp	onlinet
ups-onlinet	7010/udp	onlinet
talon-disc	7011/tcp	Talon Discovery
talon-disc	7011/udp	Talon Discovery
talon-engine	7012/tcp	Talon Engine
talon-engine	7012/udp	Talon Engine
microtalon-dis	7013/tcp	Microtalon
microtalon-dis	7013/udp	Microtalon
microtalon-com	7014/tcp	Microtalon
microtalon-com	7014/udp	Microtalon
talon-webserver	7015/tcp	Talon Webserver
talon-webserver	7015/udp	Talon Webserver
#	7016-7019	Unassigned
dpserve	7020/tcp	DP Serve
dpserve	7020/udp	DP Serve
dpserveadmin	7021/tcp	DP Serve Admin
dpserveadmin	7021/udp	DP Serve Admin
#	7022-7069	Unassigned
arcp	7070/tcp	ARCP
arcp	7070/udp	ARCP
#	7071-7098	Unassigned
lazy-ptop	7099/tcp	lazy-ptop
lazy-ptop	7099/udp	lazy-ptop
font-service	7100/tcp	X Font Service
font-service	7100/udp	X Font Service
#	7101-7120	Unassigned
virprot-lm	7121/tcp	Virtual Proto
virprot-lm	7121/udp	Virtual Proto
#	7122-7173	Unassigned
clutild	7174/tcp	Clutild
clutild	7174/udp	Clutild
#	7175-7199	Unassigned
fodms	7200/tcp	FODMS FLIP
fodms	7200/udp	FODMS FLIP
dlip	7201/tcp	DLIP
dlip	7201/udp	DLIP
swx	7300-7390	The Swiss Exch
#	7391-7394	Unassigned
winqedit	7395/tcp	winqedit
winqedit	7395/udp	winqedit
#	7396-7425	Unassigned
pmdmgr	7426/tcp	OpenView DM Post
pmdmgr	7426/udp	OpenView DM Post
oveadmgr	7427/tcp	OpenView DM Even
oveadmgr	7427/udp	OpenView DM Even
ovladmgr	7428/tcp	OpenView DM Log
ovladmgr	7428/udp	OpenView DM Log
opi-sock	7429/tcp	OpenView DM rqt
opi-sock	7429/udp	OpenView DM rqt

sd-request	2384/udp	SD-REQUEST
#	2384-2388	Unassigned
ovsessionmgr	2389/tcp	OpenView Ses Mgr
ovsessionmgr	2389/udp	OpenView Ses Mgr
rsmtp	2390/tcp	RSMTP
rsmtp	2390/udp	RSMTP
3com-net-mgmt	2391/tcp	3COM Net Mgr
3com-net-mgmt	2391/udp	3COM Net Mgr
tacticalauth	2392/tcp	Tactical Auth
tacticalauth	2392/udp	Tactical Auth
ms-olap1	2393/tcp	MS OLAP 1
ms-olap1	2393/udp	MS OLAP 1
ms-olap2	2394/tcp	MS OLAP 2
ms-olap2	2394/udp	MA OLAP 2
lan900_remote	2395/tcp	LAN900 Remote
lan900_remote	2395/udp	LAN900 Remote
wusage	2396/tcp	Wusage
wusage	2396/udp	Wusage
ncl	2397/tcp	NCL
ncl	2397/udp	NCL
orbiter	2398/tcp	Orbiter
orbiter	2398/udp	Orbiter
fmpro-fdal	2399/tcp	FileMaker, Inc.
fmpro-fdal	2399/udp	FileMaker, Inc.
opequus-server	2400/tcp	OpEquus Server
opequus-server	2400/udp	OpEquus Server
cvspserver	2401/tcp	cvspserver
cvspserver	2401/udp	cvspserver
taskmaster2000	2402/tcp	TaskMaster 2000
taskmaster2000	2402/udp	TaskMaster 2000
taskmaster2000	2403/tcp	TaskMaster 2000
taskmaster2000	2403/udp	TaskMaster 2000
iec870-5-104	2404/tcp	IEC870-5-104
iec870-5-104	2404/udp	IEC870-5-104
trc-netpoll	2405/tcp	TRC Netpoll
trc-netpoll	2405/udp	TRC Netpoll
jediserver	2406/tcp	JediServer
jediserver	2406/udp	JediServer
orion	2407/tcp	Orion
orion	2407/udp	Orion
optimanet	2408/tcp	OptimaNet
optimanet	2408/udp	OptimaNet
sns-protocol	2409/tcp	SNS Protocol
sns-protocol	2409/udp	SNS Protocol
vrts-registry	2410/tcp	VRTS Registry
vrts-registry	2410/udp	VRTS Registry
netwave-ap-mgmt	2411/tcp	Netwave AP Mgr
netwave-ap-mgmt	2411/udp	Netwave AP Mgr
cdn	2412/tcp	CDN
cdn	2412/udp	CDN
orion-rmi-reg	2413/tcp	orion-rmi-reg
orion-rmi-reg	2413/udp	orion-rmi-reg
interlingua	2414/tcp	Interlingua
interlingua	2414/udp	Interlingua
comtest	2415/tcp	COMTEST
comtest	2415/udp	COMTEST
rmtserver	2416/tcp	RMT Server
rmtserver	2416/udp	RMT Server
composit-server	2417/tcp	Composit Server
composit-server	2417/udp	Composit Server
cas	2418/tcp	cas
cas	2418/udp	cas
attachmate-s2s	2419/tcp	Attachmate S2S
attachmate-s2s	2419/udp	Attachmate S2S
dslremote-mgmt	2420/tcp	DSL Remote Mgr
dslremote-mgmt	2420/udp	DSL Remote Mgr
g-talk	2421/tcp	G-Talk
g-talk	2421/udp	G-Talk
crmsbits	2422/tcp	CRMSBITS
crmsbits	2422/udp	CRMSBITS
rnrp	2423/tcp	RNRP
xmpv7	7430/tcp	OpenView DM
xmpv7	7430/udp	OpenView DM
pmd	7431/tcp	OpenView DM
pmd	7431/udp	OpenView DM
faximum	7437/tcp	Faximum
faximum	7437/udp	Faximum
telops-lmd	7491/tcp	telops-lmd
telops-lmd	7491/udp	telops-lmd
pafec-lm	7511/tcp	pafec-lm
pafec-lm	7511/udp	pafec-lm
nta-ds	7544/tcp	FlowAnalyzer
nta-ds	7544/udp	FlowAnalyzer
nta-us	7545/tcp	FlowAnalyzer
nta-us	7545/udp	FlowAnalyzer
vsi-omega	7566/tcp	VSI Omega
vsi-omega	7566/udp	VSI Omega
#	7567-7569	Unassigned
aries-kfinder	7570/tcp	Aries Kfinder
aries-kfinder	7570/udp	Aries Kfinder
#	7571-7587	Unassigned
sun-lm	7588/tcp	Sun License Mgr
sun-lm	7588/udp	Sun License Mgr
#	7589-7632	Unassigned
pmdfmgt	7633/tcp	PMDF Management
pmdfmgt	7633/udp	PMDF Management
#	7634-7776	Unassigned
cbt	7777/tcp	cbt
cbt	7777/udp	cbt
interwise	7778/tcp	Interwise
interwise	7778/udp	Interwise
#	7779-7780	Unassigned
accu-lmgr	7781/tcp	accu-lmgr
accu-lmgr	7781/udp	accu-lmgr
#	7782-7785	Unassigned
minivend	7786/tcp	MINIVEND
minivend	7786/udp	MINIVEND
#	7787-7931	Unassigned
t2-drm	7932/tcp	Tier 2 Data
t2-drm	7932/udp	Tier 2 Data
t2-brm	7933/tcp	Tier 2 Business
t2-brm	7933/udp	Tier 2 Business
supercell	7967/tcp	Supercell
supercell	7967/udp	Supercell
#	7968-7978	Unassigned
micromuse-ncps	7979/tcp	Micromuse-ncps
micromuse-ncps	7979/udp	Micromuse-ncps
quest-vista	7980/tcp	Quest Vista
quest-vista	7980/udp	Quest Vista
#	7981-7998	Unassigned
irdmi2	7999/tcp	iRDMI2
irdmi2	7999/udp	iRDMI2
irdmi	8000/tcp	iRDMI
irdmi	8000/udp	iRDMI
vcom-tunnel	8001/tcp	VCOM Tunnel
vcom-tunnel	8001/udp	VCOM Tunnel
teradataordbms	8002/tcp	Teradata ORDBMS
teradataordbms	8002/udp	Teradata ORDBMS
#	8003-8007	Unassigned
http-alt	8008/tcp	HTTP Alternate
http-alt	8008/udp	HTTP Alternate
#	8009-8031	Unassigned
pro-ed	8032/tcp	ProEd
pro-ed	8032/udp	ProEd
mindprint	8033/tcp	MindPrint
mindprint	8033/udp	MindPrint
#	8034-8079	Unassigned
http-alt	8080/tcp	HTTP Alternate
http-alt	8080/udp	HTTP Alternate
#	8081-8129	Unassigned
indigo-vrmi	8130/tcp	INDIGO-VRMI
indigo-vrmi	8130/udp	INDIGO-VRMI

rnrp	2423/udp	RNRP
kofax-svr	2424/tcp	KOFAX-SVR
kofax-svr	2424/udp	KOFAX-SVR
fjitsuappmgr	2425/tcp	Fujitsu App Mgr
fjitsuappmgr	2425/udp	Fujitsu App Mgr
applianttcp	2426/tcp	Appliant TCP
appliantudp	2426/udp	Appliant UDP
mgcp-gateway	2427/tcp	Media Gateway
mgcp-gateway	2427/udp	Media Gateway
ott	2428/tcp	1 Way Trip Time
ott	2428/udp	1 Way Trip Time
ft-role	2429/tcp	FT-ROLE
ft-role	2429/udp	FT-ROLE
venus	2430/tcp	venus
venus	2430/udp	venus
venus-se	2431/tcp	venus-se
venus-se	2431/udp	venus-se
codasrv	2432/tcp	codasrv
codasrv	2432/udp	codasrv
codasrv-se	2433/tcp	codasrv-se
codasrv-se	2433/udp	codasrv-se
pxc-epmap	2434/tcp	pxc-epmap
pxc-epmap	2434/udp	pxc-epmap
optilogic	2435/tcp	OptiLogic
optilogic	2435/udp	OptiLogic
topx	2436/tcp	TOP/X
topx	2436/udp	TOP/X
unicontrol	2437/tcp	UniControl
unicontrol	2437/udp	UniControl
msp	2438/tcp	MSP
msp	2438/udp	MSP
sybasedbsynch	2439/tcp	SybaseDBSynch
sybasedbsynch	2439/udp	SybaseDBSynch
spearway	2440/tcp	Spearway Lockers
spearway	2440/udp	Spearway Lockser
pvsw-inet	2441/tcp	pvsw-inet
pvsw-inet	2441/udp	pvsw-inet
netangel	2442/tcp	Netangel
netangel	2442/udp	Netangel
powerclientcsf	2443/tcp	PowerClient
powerclientcsf	2443/udp	PowerClient
btpp2sectrans	2444/tcp	BT PP2 Sectrans
btpp2sectrans	2444/udp	BT PP2 Sectrans
dtn1	2445/tcp	DTN1
dtn1	2445/udp	DTN1
bues_service	2446/tcp	bues_service
bues_service	2446/udp	bues_service
ovwdb	2447/tcp	OpenView NNM
ovwdb	2447/udp	OpenView NNM
hpppssvr	2448/tcp	hpppsvr
hpppssvr	2448/udp	hpppsvr
ratl	2449/tcp	RATL
ratl	2449/udp	RATL
netadmin	2450/tcp	netadmin
netadmin	2450/udp	netadmin
netchat	2451/tcp	netchat
netchat	2451/udp	netchat
snifferclient	2452/tcp	SnifferClient
snifferclient	2452/udp	SnifferClient
madge-om	2453/tcp	madge-om
madge-om	2453/udp	madge-om
indx-dds	2454/tcp	IndX-DDS
indx-dds	2454/udp	IndX-DDS
wago-io-system	2455/tcp	WAGO-IO-SYSTEM
wago-io-system	2455/udp	WAGO-IO-SYSTEM
altav-remmgt	2456/tcp	altav-remmgt
altav-remmgt	2456/udp	altav-remmgt
rapido-ip	2457/tcp	Rapido_IP
rapido-ip	2457/udp	Rapido_IP
griffin	2458/tcp	griffin
griffin	2458/udp	griffin
indigo-vbcp	8131/tcp	INDIGO-VBCP
indigo-vbcp	8131/udp	INDIGO-VBCP
#	8132-8159	Unassigned
patrol	8160/tcp	Patrol
patrol	8160/udp	Patrol
patrol-snmp	8161/tcp	Patrol SNMP
patrol-snmp	8161/udp	Patrol SNMP
#	8162-8199	Unassigned
trivnet1	8200/tcp	TRIVNET
trivnet1	8200/udp	TRIVNET
trivnet2	8201/tcp	TRIVNET
trivnet2	8201/udp	TRIVNET
#	8202-8203	Unassigned
lm-perfworks	8204/tcp	LM Perfworks
lm-perfworks	8204/udp	LM Perfworks
lm-instmgr	8205/tcp	LM Instmgr
lm-instmgr	8205/udp	LM Instmgr
lm-dta	8206/tcp	LM Dta
lm-dta	8206/udp	LM Dta
lm-sserver	8207/tcp	LM SServer
lm-sserver	8207/udp	LM SServer
lm-webwatcher	8208/tcp	LM Webwatcher
lm-webwatcher	8208/udp	LM Webwatcher
#	8209-8350	Unassigned
server-find	8351/tcp	Server Find
server-find	8351/udp	Server Find
#	8352-8375	Unassigned
cruise-enum	8376/tcp	Cruise ENUM
cruise-enum	8376/udp	Cruise ENUM
cruise-swroute	8377/tcp	Cruise SWROUTE
cruise-swroute	8377/udp	Cruise SWROUTE
cruise-config	8378/tcp	Cruise CONFIG
cruise-config	8378/udp	Cruise CONFIG
cruise-diags	8379/tcp	Cruise DIAGS
cruise-diags	8379/udp	Cruise DIAGS
cruise-update	8380/tcp	Cruise UPDATE
cruise-update	8380/udp	Cruise UPDATE
#	8381-8399	Unassigned
cvd	8400/tcp	cvd
cvd	8400/udp	cvd
sabarsd	8401/tcp	sabarsd
sabarsd	8401/udp	sabarsd
abarsd	8402/tcp	abarsd
abarsd	8402/udp	abarsd
admind	8403/tcp	admind
admind	8403/udp	admind
#	8404-8449	Unassigned
npmp	8450/tcp	npmp
npmp	8450/udp	npmp
#	8451-8472	Unassigned
vp2p	8473/tcp	Virtual P-to-P
vp2p	8473/udp	Virtual P-to-P
#	8474-8553	Unassigned
rtsp-alt	8554/tcp	RTSP Alternate
rtsp-alt	8554/udp	RTSP Alternate
#	8555-8732	Unassigned
ibus	8733/tcp	iBus
ibus	8733/udp	iBus
#	8734-8762	Unassigned
mc-appserver	8763/tcp	MC-APPSERVER
mc-appserver	8763/udp	MC-APPSERVER
openqueue	8764/tcp	OPENQUEUE
openqueue	8764/udp	OPENQUEUE
ultraseek-http	8765/tcp	Ultraseek HTTP
ultraseek-http	8765/udp	Ultraseek HTTP
#	8766-8803	Unassigned
truecm	8804/tcp	truecm
truecm	8804/udp	truecm
#	8805-8879	
cddbp-alt	8880/tcp	CDDBP
cddbp-alt	8880/udp	CDDBP

community	2459/tcp	Community
community	2459/udp	Community
ms-theater	2460/tcp	ms-theater
ms-theater	2460/udp	ms-theater
qadmifoper	2461/tcp	qadmifoper
qadmifoper	2461/udp	qadmifoper
qadmifevent	2462/tcp	qadmifevent
qadmifevent	2462/udp	qadmifevent
symbios-raid	2463/tcp	Symbios Raid
symbios-raid	2463/udp	Symbios Raid
direcpc-si	2464/tcp	DirecPC SI
direcpc-si	2464/udp	DirecPC SI
lbm	2465/tcp	Load Balance Mgr
lbm	2465/udp	Load Balance Mgr
lbf	2466/tcp	Load Balance Fwr
lbf	2466/udp	Load Balance Fwr
high-criteria	2467/tcp	High Criteria
high-criteria	2467/udp	High Criteria
qip-msgd	2468/tcp	qip_msgd
qip-msgd	2468/udp	qip_msgd
mti-tcs-comm	2469/tcp	MTI-TCS-COMM
mti-tcs-comm	2469/udp	MTI-TCS-COMM
taskman-port	2470/tcp	taskman port
taskman-port	2470/udp	taskman port
seaodbc	2471/tcp	SeaODBC
seaodbc	2471/udp	SeaODBC
c3	2472/tcp	C3
c3	2472/udp	C3
aker-cdp	2473/tcp	Aker-cdp
aker-cdp	2473/udp	Aker-cdp
vitalanalysis	2474/tcp	Vital Analysis
vitalanalysis	2474/udp	Vital Analysis
ace-server	2475/tcp	ACE Server
ace-server	2475/udp	ACE Server
ace-svr-prop	2476/tcp	ACE Server
ace-svr-prop	2476/udp	ACE Server
ssm-cvs	2477/tcp	SecurSight
ssm-cvs	2477/udp	SecurSight
ssm-cssps	2478/tcp	SecurSight (SLL)
ssm-cssps	2478/udp	SecurSight (SSL)
ssm-els	2479/tcp	SecurSight (SSL)
ssm-els	2479/udp	SecurSight (SSL)
lingwood	2480/tcp	Lingwood's
lingwood	2480/udp	Lingwood's
giop	2481/tcp	Oracle GIOP
giop	2481/udp	Oracle GIOP
giop-ssl	2482/tcp	Oracle GIOP SSL
giop-ssl	2482/udp	Oracle GIOP SSL
ttc	2483/tcp	Oracle TTC
ttc	2483/udp	Oracel TTC
ttc-ssl	2484/tcp	Oracle TTC SSL
ttc-ssl	2484/udp	Oracle TTC SSL
netobjects1	2485/tcp	Net Objects1
netobjects1	2485/udp	Net Objects1
netobjects2	2486/tcp	Net Objects2
netobjects2	2486/udp	Net Objects2
pns	2487/tcp	Policy Notice
pns	2487/udp	Policy Notice
moy-corp	2488/tcp	Moy Corporation
moy-corp	2488/udp	Moy Corporation
tsilb	2489/tcp	TSILB
tsilb	2489/udp	TSILB
qip-qdhcp	2490/tcp	qip_qdhcp
qip-qdhcp	2490/udp	qip_qdhcp
conclave-cpp	2491/tcp	Conclave CPP
conclave-cpp	2491/udp	Conclave CPP
groove	2492/tcp	GROOVE
groove	2492/udp	GROOVE
talarian-mqs	2493/tcp	Talarian MQS
talarian-mqs	2493/udp	Talarian MQS
bmc-ar	2494/tcp	BMC AR
#	8881-8887	Unassigned
ddi-tcp-1	8888/tcp	NewsEDGE TCP 1
ddi-udp-1	8888/udp	NewsEDGE UDP 1
ddi-tcp-2	8889/tcp	TCP 1
ddi-udp-2	8889/udp	NewsEDGE server
ddi-tcp-3	8890/tcp	TCP 2
ddi-udp-3	8890/udp	NewsEDGE
ddi-tcp-4	8891/tcp	NESS app
ddi-udp-4	8891/udp	NESS app
ddi-tcp-5	8892/tcp	FARM product
ddi-udp-5	8892/udp	FARM product
ddi-tcp-6	8893/tcp	NewsEDGE
ddi-udp-6	8893/udp	NewsEDGE
ddi-tcp-7	8894/tcp	COAL app
ddi-udp-7	8894/udp	COAL app
#	8895-8899	Unassigned
jmb-cds1	8900/tcp	JMB-CDS 1
jmb-cds1	8900/udp	JMB-CDS 1
jmb-cds2	8901/tcp	JMB-CDS 2
jmb-cds2	8901/udp	JMB-CDS 2
#	8902-8999	Unassigned
cslistener	9000/tcp	CSlistener
cslistener	9000/udp	CSlistener
#	9001-9005	Unassigned
#	9006	De-Commissioned
#	9007-9089	Unassigned
websm	9090/tcp	WebSM
websm	9090/udp	WebSM
#	9091-9159	Unassigned
netlock1	9160/tcp	NetLOCK1
netlock1	9160/udp	NetLOCK1
netlock2	9161/tcp	NetLOCK2
netlock2	9161/udp	NetLOCK2
netlock3	9162/tcp	NetLOCK3
netlock3	9162/udp	NetLOCK3
netlock4	9163/tcp	NetLOCK4
netlock4	9163/udp	NetLOCK4
netlock5	9164/tcp	NetLOCK5
netlock5	9164/udp	NetLOCK5
#	9165-9199	Unassigned
wap-wsp	9200/tcp	WAP
wap-wsp	9200/udp	WAP
wap-wsp-wtp	9201/tcp	WAP session
wap-wsp-wtp	9201/udp	WAP session
wap-wsp-s	9202/tcp	WAP secure
wap-wsp-s	9202/udp	WAP secure
wap-wsp-wtp-s	9203/tcp	WAP secure
wap-wsp-wtp-s	9203/udp	WAP secure
wap-vcard	9204/tcp	WAP vCard
wap-vcard	9204/udp	WAP vCard
wap-vcal	9205/tcp	WAP vCal
wap-vcal	9205/udp	WAP vCal
wap-vcard-s	9206/tcp	WAP vCard
wap-vcard-s	9206/udp	WAP vCard
wap-vcal-s	9207/tcp	WAP vCal Secure
wap-vcal-s	9207/udp	WAP vCal Secure
#	9208-9282	Unassigned
callwaveiam	9283/tcp	CallWaveIAM
callwaveiam	9283/udp	CallWaveIAM
#	9284-9320	Unassigned
guibase	9321/tcp	guibase
guibase	9321/udp	guibase
#	9322-9342	Unassigned
mpidcmgr	9343/tcp	MpIdcMgr
mpidcmgr	9343/udp	MpIdcMgr
mphlpdmc	9344/tcp	Mphlpdmc
mphlpdmc	9344/udp	Mphlpdmc
#	9345-9373	Unassigned
fjdmimgr	9374/tcp	fjdmimgr
fjdmimgr	9374/udp	fjdmimgr
#	9375-9395	Unassigned

```
bmc-ar          2494/udp   BMC AR
fast-rem-serv   2495/tcp   Fast Remote Serv
fast-rem-serv   2495/udp   Fast Remote Serv
dirgis          2496/tcp   DIRGIS
dirgis          2496/udp   DIRGIS
quaddb          2497/tcp   Quad DB
quaddb          2497/udp   Quad DB
odn-castraq     2498/tcp   ODN-CasTraq
odn-castraq     2498/udp   ODN-CasTraq
unicontrol      2499/tcp   UniControl
unicontrol      2499/udp   UniControl
rtsserv         2500/tcp   Resource Track
rtsserv         2500/udp   Resource Track
rtsclient       2501/tcp   Resource Track
rtsclient       2501/udp   Resource Track
kentrox-prot    2502/tcp   Kentrox Protocol
kentrox-prot    2502/udp   Kentrox Protocol
nms-dpnss       2503/tcp   NMS-DPNSS
nms-dpnss       2503/udp   NMS-DPNSS
wlbs            2504/tcp   WLBS
wlbs            2504/udp   WLBS
torque-traffic  2505/tcp   torque-traffic
torque-traffic  2505/udp   torque-traffic
jbroker         2506/tcp   jbroker
jbroker         2506/udp   jbroker
spock           2507/tcp   spock
spock           2507/udp   spock
jdatastore      2508/tcp   JDataStore
jdatastore      2508/udp   JDataStore
fjmpss          2509/tcp   fjmpss
fjmpss          2509/udp   fjmpss
fjappmgrbulk    2510/tcp   fjappmgrbulk
fjappmgrbulk    2510/udp   fjappmgrbulk
metastorm       2511/tcp   Metastorm
metastorm       2511/udp   Metastorm
citrixima       2512/tcp   Citrix IMA
citrixima       2512/udp   Citrix IMA
citrixadmin     2513/tcp   Citrix ADMIN
citrixadmin     2513/udp   Citrix ADMIN
facsys-ntp      2514/tcp   Facsys NTP
facsys-ntp      2514/udp   Facsys NTP
facsys-router   2515/tcp   Facsys Router
facsys-router   2515/udp   Facsys Router
maincontrol     2516/tcp   Main Control
maincontrol     2516/udp   Main Control
call-sig-trans  2517/tcp   H.323 Annex E
call-sig-trans  2517/udp   H.323 Annex E
willy           2518/tcp   Willy
willy           2518/udp   Willy
globmsgsvc      2519/tcp   globmsgsvc
globmsgsvc      2519/udp   globmsgsvc
pvsw            2520/tcp   pvsw
pvsw            2520/udp   pvsw
adaptecmgr      2521/tcp   Adaptec Manager
adaptecmgr      2521/udp   Adaptec Manager
windb           2522/tcp   WinDb
windb           2522/udp   WinDb
qke-llc-v3      2523/tcp   Qke LLC V.3
qke-llc-v3      2523/udp   Qke LLC V.3
optiwave-lm     2524/tcp   Optiwave
optiwave-lm     2524/udp   Optiwave
ms-v-worlds     2525/tcp   MS V-Worlds
ms-v-worlds     2525/udp   MS V-Worlds
ema-sent-lm     2526/tcp   EMA License Mgr
ema-sent-lm     2526/udp   EMA License Mgr
iqserver        2527/tcp   IQ Server
iqserver        2527/udp   IQ Server
ncr_ccl         2528/tcp   NCR CCL
ncr_ccl         2528/udp   NCR CCL
utsftp          2529/tcp   UTS FTP
utsftp          2529/udp   UTS FTP
fjinvmgr        9396/tcp   fjinvmgr
fjinvmgr        9396/udp   fjinvmgr
mpidcagt        9397/tcp   MpIdcAgt
mpidcagt        9397/udp   MpIdcAgt
#               9398-9499  Unassigned
ismserver       9500/tcp   ismserver
ismserver       9500/udp   ismserver
#               9501-9534  Unassigned
man             9535/tcp
man             9535/udp
#               9536-9593  Unassigned
msgsys          9594/tcp   Message System
msgsys          9594/udp   Message System
pds             9595/tcp   Ping Discovery
pds             9595/udp   Ping Discovery
#               9596-9599  Unassigned
micromuse-ncpw  9600/tcp   MICROMUSE-NCPW
micromuse-ncpw  9600/udp   MICROMUSE-NCPW
#               9601-9752  Unassigned
rasadv          9753/tcp   rasadv
rasadv          9753/udp   rasadv
#               9754-9875  Unassigned
sd              9876/tcp   Session Direct
sd              9876/udp   Session Direct
cyborg-systems  9888/tcp   CYBORG Systems
cyborg-systems  9888/udp   CYBORG Systems
monkeycom       9898/tcp   MonkeyCom
monkeycom       9898/udp   MonkeyCom
sctp-tunneling  9899/tcp   SCTP TUNNELING
sctp-tunneling  9899/udp   SCTP TUNNELING
iua             9900/tcp   IUA
iua             9900/udp   IUA
#               9901-9908  Unassigned
domaintime      9909/tcp   domaintime
domaintime      9909/udp   domaintime
#               9910-9949  Unassigned
apcpcpluswin1   9950/tcp   APCPCPLUSWIN1
apcpcpluswin1   9950/udp   APCPCPLUSWIN1
apcpcpluswin2   9951/tcp   APCPCPLUSWIN2
apcpcpluswin2   9951/udp   APCPCPLUSWIN2
apcpcpluswin3   9952/tcp   APCPCPLUSWIN3
apcpcpluswin3   9952/udp   APCPCPLUSWIN3
#               9953-9991  Unassigned
palace          9992/tcp   Palace
palace          9992/udp   Palace
palace          9993/tcp   Palace
palace          9993/udp   Palace
palace          9994/tcp   Palace
palace          9994/udp   Palace
palace          9995/tcp   Palace
palace          9995/udp   Palace
palace          9996/tcp   Palace
palace          9996/udp   Palace
palace          9997/tcp   Palace
palace          9997/udp   Palace
distinct32      9998/tcp   Distinct32
distinct32      9998/udp   Distinct32
distinct        9999/tcp   distinct
distinct        9999/udp   distinct
ndmp            10000/tcp  Network Data
ndmp            10000/udp  Network Data
#               10001-10006 Unassigned
mvs-capacity    10007/tcp  MVS Capacity
mvs-capacity    10007/udp  MVS Capacity
#               10008-10079 Unassigned
amanda          10080/tcp  Amanda
amanda          10080/udp  Amanda
#               10081-10112 Unassigned
netiq-endpoint  10113/tcp  NetIQ Endpoint
netiq-endpoint  10113/udp  NetIQ Endpoint
netiq-qcheck    10114/tcp  NetIQ Qcheck
```

```
vrcommerce      2530/tcp   VR Commerce
vrcommerce      2530/udp   VR Commerce
ito-e-gui       2531/tcp   ITO-E GUI
ito-e-gui       2531/udp   ITO-E GUI
ovtopmd         2532/tcp   OVTOPMD
ovtopmd         2532/udp   OVTOPMD
snifferserver   2533/tcp   SnifferServer
snifferserver   2533/udp   SnifferServer
combox-web-acc  2534/tcp   Combox Web Acc
combox-web-acc  2534/udp   Combox Web Acc
madcap          2535/tcp   MADCAP
madcap          2535/udp   MADCAP
btpp2audctr1    2536/tcp   btpp2audctr1
btpp2audctr1    2536/udp   btpp2audctr1
upgrade         2537/tcp   Upgrade Protocol
upgrade         2537/udp   Upgrade Protocol
vnwk-prapi      2538/tcp   vnwk-prapi
vnwk-prapi      2538/udp   vnwk-prapi
vsiadmin        2539/tcp   VSI Admin
vsiadmin        2539/udp   VSI Admin
lonworks        2540/tcp   LonWorks
lonworks        2540/udp   LonWorks
lonworks2       2541/tcp   LonWorks2
lonworks2       2541/udp   LonWorks2
davinci         2542/tcp   daVinci
davinci         2542/udp   daVinci
reftek          2543/tcp   REFTEK
reftek          2543/udp   REFTEK
novell-zen      2544/tcp   Novell ZEN
novell-zen      2544/tcp   Novell ZEN
sis-emt         2545/tcp   sis-emt
sis-emt         2545/udp   sis-emt
vytalvaultbrtp  2546/tcp   vytalvaultbrtp
vytalvaultbrtp  2546/udp   vytalvaultbrtp
vytalvaultvsmp  2547/tcp   vytalvaultvsmp
vytalvaultvsmp  2547/udp   vytalvaultvsmp
vytalvaultpipe  2548/tcp   vytalvaultpipe
vytalvaultpipe  2548/udp   vytalvaultpipe
ipass           2549/tcp   IPASS
ipass           2549/udp   IPASS
ads             2550/tcp   ADS
ads             2550/udp   ADS
isg-uda-server  2551/tcp   ISG UDA Server
isg-uda-server  2551/udp   ISG UDA Server
call-logging    2552/tcp   Call Logging
call-logging    2552/udp   Call Logging
efidiningport   2553/tcp   efidiningport
efidiningport   2553/udp   efidiningport
vcnet-link-v10  2554/tcp   VCnet-Link v10
vcnet-link-v10  2554/udp   VCnet-Link v10
compaq-wcp      2555/tcp   Compaq WCP
compaq-wcp      2555/udp   Compaq WCP
nicetec-nmsvc   2556/tcp   nicetec-nmsvc
nicetec-nmsvc   2556/udp   nicetec-nmsvc
nicetec-mgmt    2557/tcp   nicetec-mgmt
nicetec-mgmt    2557/udp   nicetec-mgmt
pclemultimedia  2558/tcp   PCLE Multi Media
pclemultimedia  2558/udp   PCLE Multi Media
lstp            2559/tcp   LSTP
lstp            2559/udp   LSTP
labrat          2560/tcp   labrat
labrat          2560/udp   labrat
mosaixcc        2561/tcp   MosaixCC
mosaixcc        2561/udp   MosaixCC
delibo          2562/tcp   Delibo
delibo          2562/udp   Delibo
cti-redwood     2563/tcp   CTI Redwood
cti-redwood     2563/udp   CTI Redwood
hp-3000-telnet  2564/tcp   HP 3000 NS/VT
coord-svr       2565/tcp   Coordinator Serv
coord-svr       2565/udp   Coordinator Serv
netiq-qcheck    10114/udp  NetIQ Qcheck
ganymede-endpt  10115/tcp  Ganymede
ganymede-endpt  10115/udp  Ganymede
#               10116-10127 Unassigned
bmc-perf-sd     10128/tcp  BMC-PERFORM
bmc-perf-sd     10128/udp  BMC-PERFORM
#               10129-10287 Unassigned
blocks          10288/tcp  Blocks
blocks          10288/udp  Blocks
#               10289-10999 Unassigned
irisa           11000/tcp  IRISA
irisa           11000/udp  IRISA
metasys         11001/tcp  Metasys
metasys         11001/udp  Metasys
#               11002-11110 Unassigned
vce             11111/tcp  Viral (VCE)
vce             11111/udp  Viral (VCE)
#               11112-11366 Unassigned
atm-uhas        11367/tcp  ATM UHAS
atm-uhas        11367/udp  ATM UHAS
#               11368-11719 Unassigned
h323callsigalt  11720/tcp  h323 Call Signal
h323callsigalt  11720/udp  h323 Call Signal
#               11721-11999 Unassigned
entextxid       12000/tcp  IBM Enterprise
entextxid       12000/udp  IBM Enterprise
entextnetwk     12001/tcp  IBM Enterprise
entextnetwk     12001/udp  IBM Enterprise
entexthigh      12002/tcp  IBM Enterprise
entexthigh      12002/udp  IBM Enterprise
entextmed       12003/tcp  IBM Enterprise
entextmed       12003/udp  IBM Enterprise
entextlow       12004/tcp  IBM Enterprise
entextlow       12004/udp  IBM Enterprise
#               12005-12171 Unassigned
hivep           12172/tcp  HiveP
hivep           12172/udp  HiveP
#               12173-12752 Unassigned
tsaf            12753/tcp  tsaf port
tsaf            12753/udp  tsaf port
#               12754-13159 Unassigned
i-zipqd         13160/tcp  I-ZIPQD
i-zipqd         13160/udp  I-ZIPQD
#               13161-13222 Unassigned
powwow-client   13223/tcp  PowWow Client
powwow-client   13223/udp  PowWow Client
powwow-server   13224/tcp  PowWow Server
powwow-server   13224/udp  PowWow Server
#               13225-13719 Unassigned
bprd            13720/tcp  BPRD Protocol
bprd            13720/udp  BPRD Protocol
bpbrm           13721/tcp  BPBRM Protocol
bpbrm           13721/udp  BPBRM Protocol
bpjava-msvc     13722/tcp  BP Java MSVC
bpjava-msvc     13722/udp  BP Java MSVC
#               13723-13781 Unassigned
bpcd            13782/tcp  VERITAS
bpcd            13782/udp  VERITAS
vopied          13783/tcp  VOPIED Protnocol
vopied          13783/udp  VOPIED Protocol
#               13784-13817 Unassigned
dsmcc-config    13818/tcp  DSMCC Config
dsmcc-config    13818/udp  DSMCC Config
dsmcc-session   13819/tcp  DSMCC Session
dsmcc-session   13819/udp  DSMCC Session
dsmcc-passthru  13820/tcp  DSMCC Pass-Thru
dsmcc-passthru  13820/udp  DSMCC Pass-Thru
dsmcc-download  13821/tcp  DSMCC Download
dsmcc-download  13821/udp  DSMCC Download
dsmcc-ccp       13822/tcp  DSMCC Channel
dsmcc-ccp       13822/udp  DSMCC Channel
```

```
pcs-pcw          2566/tcp   pcs-pcw
pcs-pcw          2566/udp   pcs-pcw
clp              2567/tcp   Cisco Line Proto
clp              2567/udp   Cisco Line Proto
spamtrap         2568/tcp   SPAM TRAP
spamtrap         2568/udp   SPAM TRAP
sonuscallsig     2569/tcp   Sonus Call Sign
sonuscallsig     2569/udp   Sonus Call Sign
hs-port          2570/tcp   HS Port
hs-port          2570/udp   HS Port
cecsvc           2571/tcp   CECSVC
cecsvc           2571/udp   CECSVC
ibp              2572/tcp   IBP
ibp              2572/udp   IBP
trustestablish   2573/tcp   Trust Establish
trustestablish   2573/udp   Trust Establish
blockade-bpsp    2574/tcp   Blockade BPSP
blockade-bpsp    2574/udp   Blockade BPSP
hl7              2575/tcp   HL7
hl7              2575/udp   HL7
tclprodebugger   2576/tcp   TCL Pro Debugger
tclprodebugger   2576/udp   TCL Pro Debugger
scipticslsrvr    2577/tcp   Scriptics Lsrvr
scipticslsrvr    2577/udp   Scriptics Lsrvr
rvs-isdn-dcp     2578/tcp   RVS ISDN DCP
rvs-isdn-dcp     2578/udp   RVS ISDN DCP
mpfoncl          2579/tcp   mpfoncl
mpfoncl          2579/udp   mpfoncl
tributary        2580/tcp   Tributary
tributary        2580/udp   Tributary
argis-te         2581/tcp   ARGIS TE
argis-te         2581/udp   ARGIS TE
argis-ds         2582/tcp   ARGIS DS
argis-ds         2582/udp   ARGIS DS
mon              2583/tcp   MON
mon              2583/udp   MON
cyaserv          2584/tcp   cyaserv
cyaserv          2584/udp   cyaserv
netx-server      2585/tcp   NETX Server
netx-server      2585/udp   NETX Server
netx-agent       2586/tcp   NETX Agent
netx-agent       2586/udp   NETX Agent
masc             2587/tcp   MASC
masc             2587/udp   MASC
privilege        2588/tcp   Privilege
privilege        2588/udp   Privilege
quartus-tcl      2589/tcp   quartus tcl
quartus-tcl      2589/udp   quartus tcl
idotdist         2590/tcp   idotdist
idotdist         2590/udp   idotdist
maytagshuffle    2591/tcp   Maytag Shuffle
maytagshuffle    2591/udp   Maytag Shuffle
netrek           2592/tcp   netrek
netrek           2592/udp   netrek
mns-mail         2593/tcp   MNS Mail Notice
mns-mail         2593/udp   MNS Mail Notice
dts              2594/tcp   Data Base Server
dts              2594/udp   Data Base Server
worldfusion1     2595/tcp   World Fusion 1
worldfusion1     2595/udp   World Fusion 1
worldfusion2     2596/tcp   World Fusion 2
worldfusion2     2596/udp   World Fusion 2
homesteadglory   2597/tcp   Homestead Glory
homesteadglory   2597/udp   Homestead Glory
citriximaclient  2598/tcp   Citrix MA Client
citriximaclient  2598/udp   Citrix MA Client
meridiandata     2599/tcp   Meridian Data
meridiandata     2599/udp   Meridian Data
hpstgmgr         2600/tcp   HPSTGMGR
hpstgmgr         2600/udp   HPSTGMGR
discp-client     2601/tcp   discp client
#                13823-14000 Unassigned
itu-sccp-ss7     14001/tcp  ITU SCCP (SS7)
itu-sccp-ss7     14001/udp  ITU SCCP (SS7)
#                14002-16359 Unassigned
netserialext1    16360/tcp  netserialext1
netserialext1    16360/udp  netserialext1
netserialext2    16361/tcp  netserialext2
netserialext2    16361/udp  netserialext2
#                16362-16366 Unassigned
netserialext3    16367/tcp  netserialext3
netserialext3    16367/udp  netserialext3
netserialext4    16368/tcp  netserialext4
netserialext4    16368/udp  netserialext4
#                16369-16990 Unassigned
intel-rci-mp     16991/tcp  INTEL-RCI-MP
intel-rci-mp     16991/udp  INTEL-RCI-MP
#                16992-17006 Unassigned
isode-dua        17007/tcp
isode-dua        17007/udp
#                17008-17218 Unassigned
chipper          17219/tcp  Chipper
chipper          17219/udp  Chipper
#                17220-17999 Unassigned
biimenu          18000/tcp  Beckman Inc.
biimenu          18000/udp  Beckman Inc.
#                18001-18180 Unassigned
opsec-cvp        18181/tcp  OPSEC CVP
opsec-cvp        18181/udp  OPSEC CVP
opsec-ufp        18182/tcp  OPSEC UFP
opsec-ufp        18182/udp  OPSEC UFP
opsec-sam        18183/tcp  OPSEC SAM
opsec-sam        18183/udp  OPSEC SAM
opsec-lea        18184/tcp  OPSEC LEA
opsec-lea        18184/udp  OPSEC LEA
opsec-omi        18185/tcp  OPSEC OMI
opsec-omi        18185/udp  OPSEC OMI
#                18186      Unassigned
opsec-ela        18187/tcp  OPSEC ELA
opsec-ela        18187/udp  OPSEC ELA
ac-cluster       18463/tcp  AC Cluster
ac-cluster       18463/udp  AC Cluster
#                18464-18887 Unassigned
apc-necmp        18888/tcp  APCNECMP
apc-necmp        18888/udp  APCNECMP
#                18889-19190 Unassigned
opsec-uaa        19191/tcp  opsec-uaa
opsec-uaa        19191/udp  opsec-uaa
#                19192-19282 Unassigned
keysrvr          19283/tcp  Key Server
keysrvr          19283/udp  Key Server
#                19284-19314 Unassigned
keyshadow        19315/tcp  Key Shadow
keyshadow        19315/udp  Key Shadow
#                19316-19409 Unassigned
hp-sco           19410/tcp  hp-sco
hp-sco           19410/udp  hp-sco
hp-sca           19411/tcp  hp-sca
hp-sca           19411/udp  hp-sca
hp-sessmon       19412/tcp  HP-SESSMON
hp-sessmon       19412/udp  HP-SESSMON
#                19413-19540 Unassigned
jcp              19541/tcp  JCP Client
#                19542-19999 Unassigned
dnp              20000/tcp  DNP
dnp              20000/udp  DNP
#                20001-20669 Unassigned
track            20670/tcp  Track
track            20670/udp  Track
#                20671-20998 Unassigned
athand-mmp       20999/tcp  At Hand MMP
athand-mmp       20999/udp  AT Hand MMP
```

```
discp-client    2601/udp   discp client
discp-server    2602/tcp   discp server
discp-server    2602/udp   discp server
servicemeter    2603/tcp   Service Meter
servicemeter    2603/udp   Service Meter
nsc-ccs         2604/tcp   NSC CCS
nsc-ccs         2604/udp   NSC CCS
nsc-posa        2605/tcp   NSC POSA
nsc-posa        2605/udp   NSC POSA
netmon          2606/tcp   Dell Netmon
netmon          2606/udp   Dell Netmon
connection      2607/tcp   Dell Connection
connection      2607/udp   Dell Connection
wag-service     2608/tcp   Wag Service
wag-service     2608/udp   Wag Service
system-monitor  2609/tcp   System Monitor
system-monitor  2609/udp   System Monitor
versa-tek       2610/tcp   VersaTek
versa-tek       2610/udp   VersaTek
lionhead        2611/tcp   LIONHEAD
lionhead        2611/udp   LIONHEAD
qpasa-agent     2612/tcp   Qpasa Agent
qpasa-agent     2612/udp   Qpasa Agent
smntubootstrap  2613/tcp   SMNTUBootstrap
smntubootstrap  2613/udp   SMNTUBootstrap
neveroffline    2614/tcp   Never Offline
neveroffline    2614/udp   Never Offline
firepower       2615/tcp   firepower
firepower       2615/udp   firepower
appswitch-emp   2616/tcp   appswitch-emp
appswitch-emp   2616/udp   appswitch-emp
cmadmin         2617/tcp   Clinical Context
cmadmin         2617/udp   Clinical Context
priority-e-com  2618/tcp   Priority E-Com
priority-e-com  2618/udp   Priority E-Com
bruce           2619/tcp   bruce
bruce           2619/udp   bruc
lpsrecommender  2620/tcp   LPSRecommender
lpsrecommender  2620/udp   LPSRecommender
miles-apart     2621/tcp   Miles Apart
miles-apart     2621/udp   Miles Apart
metricadbc      2622/tcp   MetricaDBC
metricadbc      2622/udp   MetricaDBC
lmdp            2623/tcp   LMDP
lmdp            2623/udp   LMDP
aria            2624/tcp   Aria
aria            2624/udp   Aria
blwnkl-port     2625/tcp   Blwnkl Port
blwnkl-port     2625/udp   Blwnkl Port
gbjd816         2626/tcp   gbjd816
gbjd816         2626/udp   gbjd816
moshebeeri      2627/tcp   Moshe Beeri
moshebeeri      2627/udp   Moshe Beeri
dict            2628/tcp   DICT
dict            2628/udp   DICT
sitaraserver    2629/tcp   Sitara Server
sitaraserver    2629/udp   Sitara Server
sitaramgmt      2630/tcp   Sitara Mgrt
sitaramgmt      2630/udp   Sitara Mgr
sitaradir       2631/tcp   Sitara Dir
sitaradir       2631/udp   Sitara Dir
irdg-post       2632/tcp   IRdg Post
irdg-post       2632/udp   IRdg Post
interintelli    2633/tcp   InterIntelli
interintelli    2633/udp   InterIntelli
pk-electronics  2634/tcp   PK Electronics
pk-electronics  2634/udp   PK Electronics
backburner      2635/tcp   Back Burner
backburner      2635/udp   Back Burner
solve           2636/tcp   Solve
solve           2636/udp   Solve
#               20300-21589 Unassigned
vofr-gateway    21590/tcp  VoFR Gateway
vofr-gateway    21590/udp  VoFR Gateway
#               21591-21844 Unassigned
webphone        21845/tcp webphone
webphone        21845/udp webphone
netspeak-is     21846/tcp NetSpeak
netspeak-is     21846/udp NetSpeak
netspeak-cs     21847/tcp NetSpeak
netspeak-cs     21847/udp NetSpeak
netspeak-acd    21848/tcp NetSpeak
netspeak-acd    21848/udp NetSpeak
netspeak-cps    21849/tcp NetSpeak
netspeak-cps    21849/udp NetSpeak
#               21850-21999 Unassigned
snapenetio      22000/tcp SNAPenetIO
snapenetio      22000/udp SNAPenetIO
optocontrol     22001/tcp OptoControl
optocontrol     22001/udp OptoControl
#               22002-22272 Unassigned
wnn6            22273/tcp wnn6
wnn6            22273/udp wnn6
#               22556-22799 Unassigned
aws-brf         22800/tcp Telerate LAN
aws-brf         22800/udp Telerate LAN
#               22801-22950 Unassigned
brf-gw          22951/tcp Telerate WAN
brf-gw          22951/udp Telerate WAN
#               22952-23999 Unassigned
med-ltp         24000/tcp med-ltp
med-ltp         24000/udp med-ltp
med-fsp-rx      24001/tcp med-fsp-rx
med-fsp-rx      24001/udp med-fsp-rx
med-fsp-tx      24002/tcp med-fsp-tx
med-fsp-tx      24002/udp med-fsp-tx
med-supp        24003/tcp med-supp
med-supp        24003/udp med-supp
med-ovw         24004/tcp med-ovw
med-ovw         24004/udp med-ovw
med-ci          24005/tcp med-ci
med-net-svc     24006/tcp med-net-svc
med-net-svc     24006/udp med-net-svc
#               24007-24385 Unassigned
intel_rci       24386/tcp Intel RCI
intel_rci       24386/udp Intel RCI
#               24387-24553 Unassigned
binkp           24554/tcp BINKP
binkp           24554/udp BINKP
#               24555-24999 Unassigned
icl-twobase1    25000/tcp icl-twobase1
icl-twobase1    25000/udp icl-twobase1
icl-twobase2    25001/tcp icl-twobase2
icl-twobase2    25001/udp icl-twobase2
icl-twobase3    25002/tcp icl-twobase3
icl-twobase3    25002/udp icl-twobase3
icl-twobase4    25003/tcp icl-twobase4
icl-twobase4    25003/udp icl-twobase4
icl-twobase5    25004/tcp icl-twobase5
icl-twobase5    25004/udp icl-twobase5
icl-twobase6    25005/tcp icl-twobase6
icl-twobase6    25005/udp icl-twobase6
icl-twobase7    25006/tcp icl-twobase7
icl-twobase7    25006/udp icl-twobase7
icl-twobase8    25007/tcp icl-twobase8
icl-twobase8    25007/udp icl-twobase8
icl-twobase9    25008/tcp icl-twobase9
icl-twobase9    25008/udp icl-twobase9
icl-twobase10   25009/tcp icl-twobase10
icl-twobase10   25009/udp icl-twobase10
#               25010-25792 Unassigned
vocaltec-hos    25793/tcp  Vocaltec
```

```
imdocsvc        2637/tcp   Import Document
imdocsvc        2637/udp   Import Document
sybaseanywhere  2638/tcp   Sybase Anywhere
sybaseanywhere  2638/udp   Sybase Anywhere
aminet          2639/tcp   AMInet
aminet          2639/udp   AMInet
sai_sentlm      2640/tcp   Sabbagh
sai_sentlm      2640/udp   Sabbagh
hdl-srv         2641/tcp   HDL Server
hdl-srv         2641/udp   HDL Server
tragic          2642/tcp   Tragic
tragic          2642/udp   Tragic
gte-samp        2643/tcp   GTE-SAMP
gte-samp        2643/udp   GTE-SAMP
travsoft-ipx-t  2644/tcp   Travsoft IPX
travsoft-ipx-t  2644/udp   Travsoft IPX
novell-ipx-cmd  2645/tcp   Novell IPX CMD
novell-ipx-cmd  2645/udp   Novell IPX CMD
and-lm          2646/tcp   AND Licence Mgr
and-lm          2646/udp   AND License Mgr
syncserver      2647/tcp   SyncServer
syncserver      2647/udp   SyncServer
upsnotifyprot   2648/tcp   Upsnotifyprot
upsnotifyprot   2648/udp   Upsnotifyprot
vpsipport       2649/tcp   VPSIPPORT
vpsipport       2649/udp   VPSIPPORT
eristwoguns     2650/tcp   eristwoguns
eristwoguns     2650/udp   eristwoguns
ebinsite        2651/tcp   EBInSite
ebinsite        2651/udp   EBInSite
interpathpanel  2652/tcp   InterPathPanel
interpathpanel  2652/udp   InterPathPanel
sonus           2653/tcp   Sonus
sonus           2653/udp   Sonus
corel_vncadmin  2654/tcp   Corel VNC Admin
corel_vncadmin  2654/udp   Corel VNC Admin
unglue          2655/tcp   UNIX Nt Glue
unglue          2655/udp   UNIX Nt Glue
kana            2656/tcp   Kana
kana            2656/udp   Kana
sns-dispatcher  2657/tcp   SNS Dispatcher
sns-dispatcher  2657/udp   SNS Dispatcher
sns-admin       2658/tcp   SNS Admin
sns-admin       2658/udp   SNS Admin
sns-query       2659/tcp   SNS Query
sns-query       2659/udp   SNS Query
gcmonitor       2660/tcp   GC Monitor
gcmonitor       2660/udp   GC Monitor
olhost          2661/tcp   OLHOST
olhost          2661/udp   OLHOST
bintec-capi     2662/tcp   BinTec-CAPI
bintec-capi     2662/udp   BinTec-CAPI
bintec-tapi     2663/tcp   BinTec-TAPI
bintec-tapi     2663/udp   BinTec-TAPI
command-mq-gm   2664/tcp   Command MQ GM
command-mq-gm   2664/udp   Command MQ GM
command-mq-pm   2665/tcp   Command MQ PM
command-mq-pm   2665/udp   Command MQ PM
extensis        2666/tcp   extensis
extensis        2666/udp   extensis
vocaltec-hos    25793/udp  Vocaltec
#               25794-25999 Unassigned
quake           26000/tcp  quake
quake           26000/udp  quake
#               26001-26207 Unassigned
wnn6-ds         26208/tcp  wnn6-ds
wnn6-ds         26208/udp  wnn6-ds
#               26209-26999 Unassigned
flex-lm         27000-27009 FLEX LM (1-10)
#               27008-27998 Unassigned
tw-auth-key     27999/tcp  TW
tw-auth-key     27999/udp  Attribute
#               28000-32767 Unassigned
filenet-tms     32768/tcp  Filenet TMS
filenet-tms     32768/udp  Filenet TMS
filenet-rpc     32769/tcp  Filenet RPC
filenet-rpc     32769/udp  Filenet RPC
filenet-nch     32770/tcp  Filenet NCH
filenet-nch     32770/udp  Filenet NCH
#               32771-33433 Unassigned
traceroute      33434/tcp  traceroute use
traceroute      33434/udp  traceroute use
#               33435-36864 Unassigned
kastenxpipe     36865/tcp  KastenX Pipe
kastenxpipe     36865/udp  KastenX Pipe
#               36866-40840 Unassigned
cscp            40841/tcp  CSCP
cscp            40841/udp  CSCP
#               40842-43187 Unassigned
rockwell-encap  44818/tcp  Rockwell Encaps
rockwell-encap  44818/udp  Rockwell Encaps
#               44819-45677 Unassigned
eba             45678/tcp  EBA PRISE
eba             45678/udp  EBA PRISE
#               45679-45965 Unassigned
ssr-servermgr   45966/tcp  SSRServerMgr
ssr-servermgr   45966/udp  SSRServerMgr
#               45967-47556 Unassigned
dbbrowse        47557/tcp  Databeam Corp
dbbrowse        47557/udp  Databeam Corpo
#               47558-47623 Unassigned
directplaysrvr  47624/tcp  Direct Play Serv
directplaysrvr  47624/udp  Direct Play Serv
#               47625-47805 Unassigned
ap              47806/tcp  ALC Protocol
ap              47806/udp  ALC Protocol
#               47807      Unassigned
bacnet          47808/tcp  Building Aut
bacnet          47808/udp  Building Aut
#               47809-47999 Unassigned
nimcontroller   48000/tcp  Nimbus Control
nimcontroller   48000/udp  Nimbus Control
nimspooler      48001/tcp  Nimbus Spooler
nimspooler      48001/udp  Nimbus Spooler
nimhub          48002/tcp  Nimbus Hub
nimhub          48002/udp  Nimbus Hub
nimgtw          48003/tcp  Nimbus Gateway
nimgtw          48003/udp  Nimbus Gateway
#               48004-49151 Unassigne
```

Trojan Ports:

This is a list of ports commonly used by **Trojan horses**. Please note that all ports are TCP unless UDP is stated.

```
Decimal Trojan(s)
------- ---------

    2  - Death
    21 - Back Construction, Blade Runner, Doly Trojan, Fore, FTP trojan, Invisible
    FTP, Larva, MBT, Motiv, Net Administrator, Senna Spy FTP Server, WebEx, WinCrash
    23 - Tiny Telnet Server, Truva Atl
    25 - Aji, Antigen, Email Password Sender, Gip, Happy 99, I Love You, Kuang 2,
    Magic Horse, Moscow Email Trojan, Naebi, NewApt, ProMail trojan, Shtrilitz,
    Stealth, Tapiras, Terminator, WinPC, WinSpy
    31 - Agent 31, Hackers Paradise, Masters Paradise
    41 - DeepThroat
    48 - DRAT
    50 - DRAT
    59 - DMSetup
    79 - Firehotcker
    80 - Back End, Executor, Hooker, RingZero
    99 - Hidden Port
    110 - ProMail trojan
    113 - Invisible Identd Deamon, Kazimas
    119 - Happy 99
    121 - JammerKillah
    123 - Net Controller
    133 - Farnaz, port 146 - Infector
    146 - Infector(UDP)
    170 - A-trojan
    421 - TCP Wrappers
    456 - Hackers Paradise
    531 - Rasmin
    555 - Ini-Killer, NeTAdministrator, Phase Zero, Stealth Spy
    606 - Secret Service
    666 - Attack FTP, Back Construction, NokNok, Cain & Abel, Satanz Backdoor,
    ServeU, Shadow Phyre
    667 - SniperNet
    669 - DP Trojan
    692 - GayOL
    777 - Aim Spy
    808 - WinHole
    911 - Dark Shadow
    999 - DeepThroat, WinSatan
    1000 - Der Spacher 3
    1001 - Der Spacher 3, Le Guardien, Silencer, WebEx
    1010 - Doly Trojan
    1011 - Doly Trojan
    1012 - Doly Trojan
    1015 - Doly Trojan
    1016 - Doly Trojan
    1020 - Vampire
    1024 - NetSpy
    1042 - Bla
    1045 - Rasmin
    1050 - MiniCommand
    1080 - WinHole
    1081 - WinHole
    1082 - WinHole
    1083 - WinHole
    1090 - Xtreme
    1095 - RAT
    1097 - RAT
    1098 - RAT
    1099 - BFevolution, RAT
    1170 - Psyber Stream Server, Streaming Audio trojan, Voice
    1200 - NoBackO (UDP)
    1201 - NoBackO (UDP)
    1207 - SoftWAR
    1212 - Kaos
```

```
1225 - Scarab
1234 - Ultors Trojan
1243 - BackDoor-G, SubSeven, SubSeven Apocalypse, Tiles
1245 - VooDoo Doll
1255 - Scarab
1256 - Project nEXT
1269 - Mavericks Matrix
1313 - NETrojan
1338 - Millenium Worm
1349 - BO DLL (UDP)
1492 - FTP99CMP
1509 - Psyber Streaming Server
1524 - Trinoo
1600 - Shivka-Burka
1777 - Scarab
1807 - SpySender
1966 - Fake FTP
1969 - OpC BO
1981 - Shockrave
1999 - BackDoor, TransScout
2000 - Der Spaeher 3, Insane Network, TransScout
2001 - Der Spaeher 3, TransScout, Trojan Cow
2002 - TransScout
2003 - TransScout
2004 - TransScout
2005 - TransScout
2023 - Ripper
2080 - WinHole
2115 - Bugs
2140 - Deep Throat, The Invasor
2155 - Illusion Mailer
2283 - HVL Rat5
2300 - Xplorer
2565 - Striker
2583 - WinCrash
2600 - Digital RootBeer
2716 - The Prayer
2773 - SubSeven
2801 - Phineas Phucker
3000 - Remote Shutdown
3024 - WinCrash
3128 - RingZero
3129 - Masters Paradise
3150 - Deep Throat, The Invasor
3456 - Teror Trojan
3459 - Eclipse 2000, Sanctuary
3700 - Portal of Doom
3791 - Eclypse
3801 - Eclypse (UDP)
4000 - Skydance
4092 - WinCrash
4242 - Virtual hacking Machine
4321 - BoBo
4444 - Prosiak, Swift remote
4567 - File Nail
4590 - ICQTrojan
5000 - Bubbel, Back Door Setup, Sockets de Troie
5001 - Back Door Setup, Sockets de Troie
5010 - Solo
5011 - One of the Last Trojans (OOTLT)
5031 - NetMetropolitan
5031 - NetMetropolitan
5321 - Firehotcker
5343 - wCrat
5400 - Blade Runner, Back Construction
5401 - Blade Runner, Back Construction
5402 - Blade Runner, Back Construction
5550 - Xtcp
5512 - Illusion Mailer
5555 - ServeMe
5556 - BO Facil
```

```
5557 - BO Facil
5569 - Robo-Hack
5637 - PC Crasher
5638 - PC Crasher
5742 - WinCrash
5882 - Y3K RAT (UDP)
5888 - Y3K RAT
6000 - The Thing
6006 - The Thing
6272 - Secret Service
6400 - The Thing
6667 - Schedule Agent
6669 - Host Control, Vampyre
6670 - DeepThroat, BackWeb Server, WinNuke eXtreame
6711 - SubSeven
6712 - Funny Trojan, SubSeven
6713 - SubSeven
6723 - Mstream
6771 - DeepThroat
6776 - 2000 Cracks, BackDoor-G, SubSeven
6838 - Mstream (UDP)
6912 - Shit Heep (not port 69123!)
6939 - Indoctrination
6969 - GateCrasher, Priority, IRC 3, NetController
6970 - GateCrasher
7000 - Remote Grab, Kazimas, SubSeven
7001 - Freak88
7215 - SubSeven
7300 - NetMonitor
7301 - NetMonitor
7306 - NetMonitor
7307 - NetMonitor
7308 - NetMonitor
7424 - Host Control
7424 - Host Control (UDP)
7789 - Back Door Setup, ICKiller
7983 - Mstream
8080 - RingZero
8787 - Back Orifice 2000
8897 - HackOffice
8988 - BacHack
8989 - Rcon
9000 - Netministrator
9325 - Mstream (UDP)
9400 - InCommand
9872 - Portal of Doom
9873 - Portal of Doom
9874 - Portal of Doom
9875 - Portal of Doom
9876 - Cyber Attacker, RUX
9878 - TransScout
9989 - iNi-Killer
9999 - The Prayer
10067 - Portal of Doom (UDP)
10085 - Syphillis
10086 - Syphillis
10101 - BrainSpy
10167 - Portal of Doom (UDP)
10528 - Host Control
10520 - Acid Shivers
10607 - Coma
10666 - Ambush (UDP)
11000 - Senna Spy
11050 - Host Control
11051 - Host Control
11223 - Progenic trojan, Secret Agent
12076 - Gjamer
12223 - Hack´99 KeyLogger
12345 - GabanBus, My Pics, NetBus, Pie Bill Gates, Whack Job, X-bill
12346 - GabanBus, NetBus, X-bill
12349 - BioNet
```

```
12361 - Whack-a-mole
12362 - Whack-a-mole
12623 - DUN Control (UDP)
12624 - Buttman
12631 - WhackJob
12754 - Mstream
13000 - Senna Spy
13010 - Hacker Brazil
15092 - Host Control
15104 - Mstream
16660 - Stacheldracht
16484 - Mosucker
16772 - ICQ Revenge
16969 - Priority
17166 - Mosaic
17300 - Kuang2 The Virus
17777 - Nephron
18753 - Shaft (UDP)
19864 - ICQ Revenge
20001 - Millennium
20002 - AcidkoR
20034 - NetBus 2 Pro, NetRex, Whack Job
20203 - Chupacabra
20331 - Bla
20432 - Shaft
20432 - Shaft (UDP)
21544 - GirlFriend, Kidterror, Schwindler, WinSp00fer
22222 - Prosiak
23023 - Logged
23432 - Asylum
23456 - Evil FTP, Ugly FTP, Whack Job
23476 - Donald Dick
23476 - Donald Dick (UDP)
23477 - Donald Dick
26274 - Delta Source (UDP)
26681 - Spy Voice
27374 - SubSeven
27444 - Trinoo (UDP)
27573 - SubSeven
27665 - Trinoo
29104 - Host Control
29891 - The Unexplained (UDP)
30001 - TerrOr32
30029 - AOL Trojan
30100 - NetSphere
30101 - NetSphere
30102 - NetSphere
30103 - NetSphere
30103 - NetSphere (UDP)
30133 - NetSphere
30303 - Sockets de Troie
30947 - Intruse
30999 - Kuang2
31335 - Trinoo (UDP)
31336 - Bo Whack, ButtFunnel
31337 - ["ELEET" port] - Baron Night, BO client, BO2, Bo Facil
31337 - ["ELEET" port] - BackFire, Back Orifice, DeepBO, Freak> (UDP)
31338 - NetSpy DK, ButtFunnel
31338 - Back Orifice, DeepBO (UDP)
31339 - NetSpy DK
31666 - BOWhack
31785 - Hack´a´Tack
31787 - Hack´a´Tack
31788 - Hack´a´Tack
31789 - Hack´a´Tack (UDP)
31791 - Hack´a´Tack (UDP)
31792 - Hack´a´Tack
32100 - Peanut Brittle, Project nEXT
32418 - Acid Battery
33333 - Blakharaz, Prosiak
33577 - PsychWard
```

```
33777 - PsychWard
33911 - Spirit 2001a
34324 - BigGluck, TN
34555 - Trinoo (Windows) (UDP)
35555 - Trinoo (Windows) (UDP)
37651 - YAT
40412 - The Spy
40421 - Agent 40421, Masters Paradise
40422 - Masters Paradise
40423 - Masters Paradise
40426 - Masters Paradise
41666 - Remote Boot
41666 - Remote Boot (UDP)
44444 - Prosiak
47262 - Delta Source (UDP)
50505 - Sockets de Troie
50766 - Fore, Schwindler
51996 - Cafeini
52317 - Acid Battery 2000
53001 - Remote Windows Shutdown
54283 - SubSeven
54320 - Back Orifice 2000
54321 - School Bus
54321 - Back Orifice 2000 (UDP)
57341 - NetRaider
58339 - ButtFunnel
60000 - Deep Throat
60068 - Xzip 6000068
60411 - Connection
61348 - Bunker-Hill
61466 - Telecommando
61603 - Bunker-Hill
63485 - Bunker-Hill
65000 - Devil, Stacheldracht
65432 - The Traitor
65432 - The Traitor (UDP)
65535 - RC
```